Introducing Loading Uncertainty in Level Set-Based Structural Topology Optimisation

Dunning, Peter

Award date:
2011

Awarding institution:
University of Bath

Introducing Loading Uncertainty in Level Set-Based Structural Topology Optimisation

submitted by

Peter Donald Dunning

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mechanical Engineering

May 2011

Abstract

Structural topology optimisation aims to provide solutions that are independent of the initial layout, allowing the greatest opportunity to find the best design. Recent developments have seen increased interest in level set based optimisation, as the solutions obtained possess smooth boundaries and are free of numerical instabilities that affect traditional element based methods. The direct level set based structural topology optimisation method has been successful in solving a range of problems. However, it does suffer from some drawbacks and limitations.

Two of the main issues with the direct level set method are accurate and efficient sensitivity computation on the boundary and initial design dependent solutions for two dimensional problems. These issues are addressed in this thesis by investigating and improving the efficient area-weighted fixed grid method and creating a novel hole creation method for two dimensional problems based on a pseudo third dimension.

Uncertainty is important to include during design and optimisation to produce structures that are reliable and robust. Loading magnitude and direction uncertainty is introduced into the minimisation of compliance problem by considering the robust expected compliance objective. An efficient formulation is derived using an analytical approach where uncertainties are normally distributed. The robust problem is then extended to include compliance variance and sensitivities are derived using the adjoint method.

Acknowledgements

I begin by thanking my supervisors Dr. Alicia Kim and Professor Glen Mullineux for their advice, expertise and support.

I would also like to thank my fellow PhD students for their helpful discussions and suggestions, particularly Dave Betts, Phil Browne, Tom Makin, Vincent Seow, Chris Brampton and Dr. Caroline Edwards.

I have presented work in this thesis at several conferences and wish to thank numerous people I met for their helpful questions and suggestions.

I thank the Numerical Analysis Group at the Rutherford Appleton Laboratory for use of their FORTRAN HSL packages.

This research was undertaken as part of the Innovative Design and Manufacturing Research Centre (IdMRC) at the University of Bath, funded by the Engineering and Physical Sciences Research Council (EPSRC) and their support is gratefully acknowledged.

Contents

List of figures

List of tables

Nomenclature

Latin Symbols

a_i	Area of sub-domain i
b	Body forces
B	Strain displacement matrix
c_i	Polynomial co-efficient
C	Compliance
D^j	Finite difference operator in direction j
E	Young's Modulus
$\tilde{E}$	Elasticity tensor or matrix
$E[C]$	Expected value of compliance
f	Surface tractions
$\{f\}$	Vector of nodal loads
f_i	i^{th} entry in load vector
h	Element edge length
$\overline{h}$	Artificial height
$H(\phi)$	Heaviside step function
K_A	Stiffness matrix of an approximated element
K_I	Stiffness matrix of an element completely within the structure
K_{P4}	Stiffness matrix of five node bi-linear element
$[K]$	Stiffness matrix

n	Unit normal vector to the boundary
N_i	Shape function
p	Solution to the adjoint equation
$P(x)$	Probability density function of x
t	Time
u	Displacement field
u_0	Prescribed displacements
$\{u\}$	Vector of nodal displacements
U	Kinematically admissible displacements
v	Any admissible displacement field
$Var[C]$	Variance of compliance
V_n	Normal velocity function
V_{ext}	Extended velocity function
Vol^*	Volume limit
$\overline{Vol}$	Upper limit on material removal due to hole insertion
w	Weight factor

Greek Symbols

α	Area-fraction
β	Factor for CFL condition
γ	Positive small number
Γ_D	Part of Γ_S subject to zero displacement boundary conditions
Γ_F	Part of Γ_S subject to surface tractions
Γ_N	Part of Γ_S not subject to displacement boundary conditions
Γ_S	Structural boundary
Γ_0	Free part of Γ_S
Δt	Time step
ε	Strain tensor

η	Combined objective weight
θ	Loading direction
Θ	Function of $\kappa_{i,j}$ and θ
$\kappa_{i,j}$	Entry in the inverse stiffness matrix
$\bar{\kappa}$	Mean curvature
λ	Lagrange multiplier
$\bar{\lambda}$	Lagrange multiplier for secondary implicit function
μ	Mean loading magnitude
μ_θ	Mean loading direction
ν	Poisson's ratio
ρ	Shape parameter for half normal distribution
σ	Standard deviation of loading magnitude
σ_θ	Standard deviation of loading direction
$[\bar{\sigma}]$	Pseudo co-variance matrix
ς	Shape sensitivity
ϕ	Implicit level set function
$\bar{\phi}$	Secondary implicit function
ϕ_t	Temporary signed distance implicit function
ω	Narrow band width
Ω	Design domain
Ω_e	Element domain
Ω_S	Structural domain

Chapter 1

Introduction

1.1 Engineering optimisation

In a world of limited resources, maximising efficiency and reducing waste is an important and constant challenge. This is especially true in the discipline of engineering, where the goal is often to find the best solution to a practical problem. Hence optimisation lies at the core of engineering design. Traditionally the engineering design process is a manual one, involving analysis, design update and re-analysis until a sufficient design is reached. This process is often heuristic involving expert input based on knowledge and experience and may not arrive at the absolute best solution. Ever advancing improvements in computer hardware and software allow more of the engineering design process to be automated. This provides a designer with more opportunity to explore design alternatives, increasing the chance of arriving at the best solution.

1.2 Structural optimisation

The role of structural optimisation in engineering design is to provide the best possible structure that meets all prescribed constraints. For example the best structure for an aerospace design might be the one with the lowest mass that meets all stress, fatigue and manufacturing constraints. Therefore, a structural optimisation problem is often stated as finding the extremum of a cost function subject to a set of constraints. There are different types of structural optimisation that can be defined by the level of freedom an optimisation process has to change the initial design.

Sizing optimisation can be performed on structures where the layout and major dimensions have been determined and only finer sizing details are to be determined, Figure 1-1a. Shape optimisation allows for greater freedom as only the general layout has been set, but positions and sizes of features are to be determined, Figure 1-1b. Topology optimisation allows the greatest freedom, as both the layout and sizing of the structure are to be determined, Figure 1-1c.

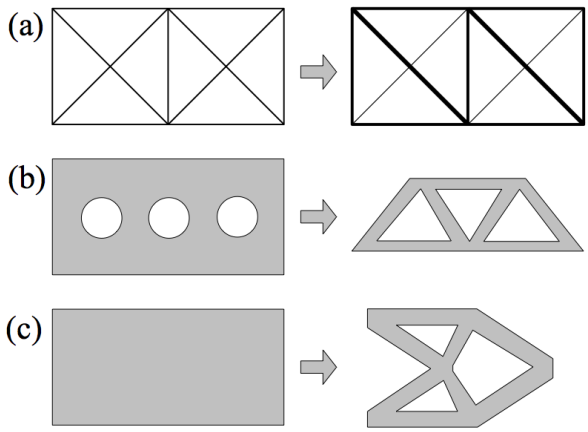

Figure 1-1: Levels of structural optimisation. a) Sizing optimisation, b) Shape optimisation, c) Topology optimisation.

Topology optimisation is particularly attractive, as its freedom allows the greatest opportunity to find the best possible design. However, this freedom usually translates into a large number of design variables that can make topology optimisation problems difficult to solve. Since computers became powerful and cheap enough to analyse structures of significant complexity, much research has been undertaken into developing computational methods for solving structural topology optimisation problems for a wide range of applications.

1.3 Importance of uncertainty

Despite the potential of topology optimisation, it has rarely been applied to design practical structures for industrial applications. One possible explanation is that a structure is optimised for one specific condition, whereas in reality the structure will operate in a range of conditions. When a range of conditions are considered, the performance of the optimised structure can be significantly lower or non-optimal. The range of operating conditions can be thought of as an uncertain environment, where there can be variability in parameters such as loading, geometry, or material properties. Therefore, including uncertainty in structural topology optimisation is important to produce practical designs.

The safety factor is the traditional engineering approach to manage uncertainty. However, safety factors are often heuristically defined and do not take direct account of uncertainties. This can lead to an overly conservative design or an under designed structure that fails in the uncertain environment. Therefore, there has been a lot of interest in using probabilistic methods that take a more direct account of uncertainties to protect against failure and reduce conservatism. However, the probabilistic paradigm has only recently been applied to topology optimisation and current methods can be inefficient and limited.

1.4 Thesis overview

The work in this thesis is concerned with introducing uncertainty in structural topology optimisation. The problem studied is the minimisation of total strain energy or compliance subject to static equilibrium and an upper limit on the available material:

$$\begin{aligned} &\text{Minimise: } C(u)=\int_{\Omega_S} \tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(u)_{kl}\,d\,\Omega_S \\ &\text{Subject to: } \int_{\Omega_S} d\,\Omega_S \le Vol^* \\ &\qquad \int_{\Omega_S} \tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(v)_{kl}\,d\,\Omega_S=\int_{\Omega_S} b\,v\,d\,\Omega_S+\int_{\Gamma_S} f\,v\,d\,\Gamma_S \\ &\qquad u|_{\Gamma_D}=u_0 \quad \forall\, v\in U \end{aligned} \tag{1.1}$$

where Ω_S is the domain of the structure, Γ_S the boundary of the structure, Vol^* the limit on material volume, $\tilde{E}_{ijkl}$ the elasticity tensor, $\varepsilon(u)_{ij}$ the strain tensor under displacement field u in the space U of kinematically permissible displacement fields subject to prescribed displacements u_o along part of the boundary Γ_D, b are body forces and f are surface tractions. Displacements and forces are vectors with directional components equal to the dimensionality of the structure. For example two directional components are required for two dimensional structures.

Minimising the compliance of a structure is equivalent to maximising its stiffness under the loading conditions. It is also equivalent to minimising the internal strain energy or the work done by the surface tractions and body forces. The upper limit on material volume makes the compliance problem (1.1) one of optimal material distribution. The structures produced by solving (1.1) provide an optimal arrangement of the available material in order to minimise structural compliance or the work done by the loading conditions. This approach produces optimal structures in the sense that all of the available material is utilised in a useful manner by supporting the loading conditions. This is a useful concept, especially in the early design stage, when the general layout of the structure is yet to be determined.

The compliance problem has been widely studied in the context of structural topology optimisation. In this work, uncertainties in surface traction loading conditions are introduced into the compliance problem (1.1) to produce robust structures that are insensitive to the uncertainties.

The compliance problem with uncertain loading conditions is studied within the following scope:

- Two dimensional linear elastic structures.
- Level set based structural topology optimisation method .
- Uncertainty in loading magnitude and direction.
- Uncertainties assumed statistically independent and normally distributed.

Within this scope, the key contributions of this thesis are:

- Investigation and development of fixed grid finite element methods that are suitable for use with the level set method optimisation method, (Chapters 3 and 4).
- Implementation of a level set optimisation method to solve the compliance problem (1.1), including the creation of new hole insertion method, (Chapters 5 and 6).
- Development of an efficient approach to solve the compliance problem under loading uncertainties to produce robust structures. Both expected compliance and compliance variance are considered, (Chapters 7 and 8).

Chapter 2

Background and literature review

2.1 Introduction

The purpose of this chapter is to provide background to the methods and ideas used throughout this thesis and to critically review the relevant literature. The review is composed of four main parts. The first part provides an overview of the development of structural topology optimisation and the various approaches. The second part reviews the level set method for topology optimisation in more detail with current trends and developments. The third part assesses various methods that have been employed to include uncertainty in structural optimisation, with particular focus on topology optimisation. The final part briefly reviews optimisation methods. The chapter concludes with some remarks on the literature that are used to motivate the research presented in this thesis.

2.2 Structural topology optimisation

Topology optimisation is the most general type of structural optimisation where the layout, shape and some sizing of the structure is considered. This allows the greatest freedom in finding the best possible design as the solution can be independent of the initial structure. For practical problems, the freedom of topology optimisation usually translates into a large number of design variables. Therefore, computational methods are usually employed to solve practical structural topology optimisation problems. Numerous methods exist and the following sections provide an overview of their development and history.

2.2.1 Truss based methods

Structural topology optimisation can be traced back to the design of truss type structures, pioneered by Michell (1904). The early approach was analytical and limited to reasonably simple structures and computational methods were developed for more practical and complex structures. Computational truss topology optimisation methods usually start with a ground structure that consists of a fixed number of nodes with fixed locations connected with a large number of initial members (Bendsøe & Sigmund 2004). The behaviour of the truss structure is usually simplified by assuming the cross sectional area of each member is small compared to its length and thus the member can be considered one dimensional. Also, the effect of member intersections on strength, stiffness and weight is usually ignored.

Design variables are then the cross sectional areas of each member in the ground structure, which can vary between prescribed upper and lower limits. The lower limit is often a small positive value to avoid numerical problems, such as stiffness matrix singularity, when analysing the structure using the Finite Element

Method (FEM). An iterative process is applied to change the design variables until an optimum solution is found. This leads to an optimum topology and also some limited shape and sizing optimisation, due to the variation of member cross sectional area. In layout optimisation the optimised topology is subject to subsequent geometrical optimisation of nodal (or joint) co-ordinates and detailed sizing optimisation of member cross sections. However, the topology may become non-optimum when considered after the subsequent geometry and sizing optimisations (Achtziger 2007).

2.2.2 Element based methods

Truss topology optimisation has been successful in solving a wide range of structural optimisation problems. However, the method can only produce truss type structures and is limited by the nodal locations and complexity of the ground structure. These limitations have seen the extension of topology optimisation to continuum type structures, which are not as limited by nodal locations and can produce more general types of structure. Research in this area started with element based methods, where the design space is discretized into a finite number of elements. The design of the structure can be changed and optimised by completely filling or removing material from an element in the domain. Thus, elements either have material or are void and the design variables are the existence of material within each element in the design space. However, this problem is ill-posed and difficult to solve because the of the on-off nature of the design variables (Strang & Kohn 1986). Therefore, the problem is usually relaxed to allow a continuous variation of material within each element. For example, the amount of material within an element for a two dimensional problem can be conceptualised as the thickness of a sheet in a panel structure (Rossow & Taylor 1973).

Microstructure and homogenisation

A different approach to manage the variable amount of material within an element is to use microstructure, where each element represents a specific geometry on a scale much smaller than the element itself (Bendsøe & Kikuchi 1988; Bendsøe 1989; Suzuki & Kikuchi 1991). The design variables are then the geometry of the microstructure within each element, which is linked to the amount of material. The macroscopic properties of each element with microstructure are approximated using a homogenisation method. This approach was an important advancement in structural topology optimisation, as physical meaning was introduced to the variable amount of material within each element. However, the use of microstructure introduces multiple design variables for each element, making the problem more computationally expensive to solve.

SIMP

A simpler method to relax the on/off ill-posed problem is to treat the amount of material within each element as a density value (Bendsøe 1989; Rozvany 2001). Design variables are the element densities and the number of variables is reduced compared to the homogenisation method. The stiffness of each element is simply scaled proportionally to its density value, therefore homogenisation techniques are not required, which also improves efficiency. However, the meaning of intermediate densities for a continuous structure can be unclear. Therefore, density values are usually penalised by raising them to a power greater than one. This effectively forces solutions towards an on/off type structure, although intermediate densities can still exist in the solution. The density approach is commonly known as SIMP (Simple Isotropic Material with Penalisation) and has been successful in efficiently solving a wide range of structural topology optimisation problems (Bendsøe & Sigmund 2004).

Despite the success of the SIMP method, it has some notable drawbacks. Firstly solutions often contain intermediate densities around the structure

boundary, Figure 2-1a. These “fuzzy boundaries” can make it difficult to interpret solutions as continuous solid structures, without using heuristic post processing methods (Tang & Chang 2001; Hsu et al. 2001). There are also two main numerical issues with the SIMP method, mesh dependent solutions and checkerboard formations in solutions (Sigmund & Petersson 1998). The latter is also a problem for the homogenisation approach.

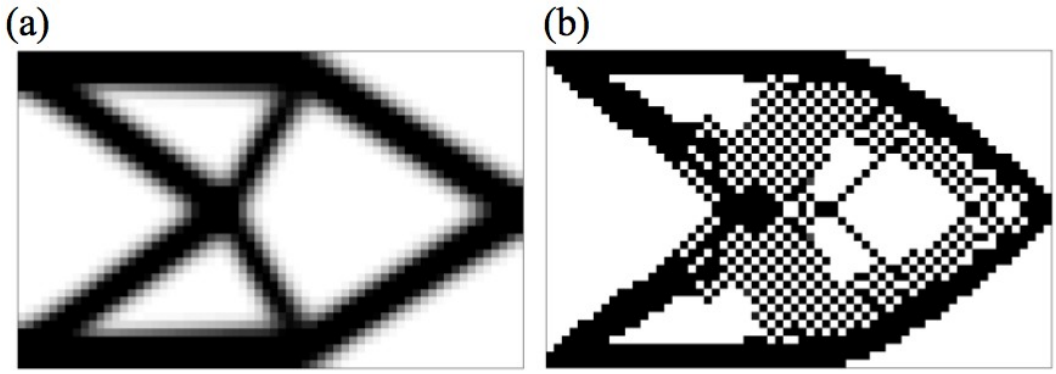

Figure 2-1: Short cantilever. a) Solution with "fuzzy" boundary, b) Solution containing checkerboard patterns.

Checkerboard patterns are regions where elements alternate between a high and low amount of material, Figure 2-1b. These patterns are regarded as numerical artefacts and not realistic optimum arrangements of material (Diaz & Sigmund 1995; Jog & Haber 1996). Solutions to this problem include introducing a constraint on the structure perimeter into the problem, although solutions can be dependent on the constraint, which can also be difficult to compute (Haber et al. 1996; Sigmund & Petersson 1998). It has also been suggested that using higher order finite elements can alleviate the problem, although this increases computational cost and does not guarantee prevention of checkerboard formation (Diaz & Sigmund 1995; Jog & Haber 1996). The most successful method is to use a filtering technique from image processing to smooth out the element sensitivity values (Sigmund & Petersson 1998). Although heuristic, the sensitivity filtering technique is simple to implement and does not significantly increase computational cost or introduce additional constraints.

Mesh dependency of solutions refers to qualitatively different solutions obtained when using different numbers of elements to discretize the problem. This problem is not suffered by the homogenisation approach because of the presence of microstructure (Sigmund & Petersson 1998). The perimeter constraint method can also be used to alleviate this problem, although the solution can become dependent on the additional constraint instead. An alternative approach is to constrain the local density variation by imposing a point-wise constraint on the derivatives of the density function (Petersson & Sigmund 1998). This method was shown to remove or sufficiently weaken numerical anomalies. However, a significant number of extra constraints are introduced, increasing computational cost.

A simpler method, involving no additional constraints, is to extend the sensitivity filtering method to be mesh independent. Although heuristic, it has been suggested that this approach provides very similar solutions to those using the density variation constraint method (Sigmund & Petersson 1998). As the filter method also suppresses checkerboard patterns the mesh independent filter method is now widely used to obtain good solutions with the SIMP method. It is also noted that all the methods for obtaining mesh independent solutions discussed here also aid in suppressing checkerboard patterns (Sigmund & Petersson 1998).

Evolutionary methods

There also exists another class of element based approaches called evolutionary methods, where topology optimisation is performed using some heuristic criteria. Examples of these methods are the Soft Kill Option inspired by adaptive bone mineralization (Baumgartner et al. 1992), where the Young's modulus is modified according to stress distribution, and Evolutionary Structural Optimisation (ESO) (Xie & Steven 1997), where low stressed material is progressively removed. ESO uses heuristic methods to obtain an optimum solution, although it often produces solutions similar to other element based methods (Xie & Steven 1997). ESO can

be applied to optimise for numerous structural objectives and constraints, such as stiffness, natural frequency and buckling (Q. Li et al. 2001). An important advancement of ESO was to introduce a mechanism to add material that can be combined with material removal to produce a Bi-directional ESO (BESO) (Querin et al. 2000, Huang et al. 2007). The advantages of BESO are faster convergence and a more robust method, as material can be both added and removed.

ESO produces clear solutions without intermediate element densities. However, it still suffers from some of the same numerical problems, such as checkerboard pattern formation (Q. Li et al. 2001). Furthermore, the boundaries obtained from ESO methods tend to be jagged and, although they are not "fuzzy", they can still as difficult to interpret as realistic structures (Hinton & Sienz, 1995). Jagged boundaries emerge because entire elements are removed during optimisation. Smoother boundaries can be obtained by allowing partial elements to be removed (H. Kim et al. 2000).

2.2.3 Boundary based methods

Element based topology optimisation methods have been successfully applied to solve many problems for continuum structures. However, the approach can suffer from numerical problems, such as checkerboard formation, and solutions often possess unclear "fuzzy" boundaries. These limitations have led to the development of boundary based topology optimisation methods, where the design variable is the location of the boundary. This paradigm inherently eliminates unclear boundaries and checkerboard patterns are not observed in solutions (S. Lee et al. 2007; K. Park & Youn 2008).

Boundary based topology optimisation methods are sometimes extensions of shape optimisation methods to include the facility to alter the topology. The topology can be changed by splitting or merging existing holes and introducing

new holes. The bubble method is an early example combining shape optimisation with the creation of new holes (Eschenauer et al. 1994). The approach was first to optimise the structure purely for shape, then use a criterion to find the optimal location to insert a new hole and therefore change the topology. The new structure is again optimised for shape producing an alternative optimal design. This process can be continually repeated producing a set of optimal designs for different topologies, from which the best design can be selected. However, this approach is inefficient as a topology optimisation method because it does not optimise both shape and topology simultaneously.

The bubble method uses splines to describe the boundary and the positions of the spline control points naturally become the design variables. Splines are often used in computer graphics as an efficient and flexible way to describe boundaries or surfaces (Tang & Chang 2001). Therefore, splines seem like a natural choice for boundary based structural topology optimisation because solutions can be directly transferred into computer aided design (CAD) packages, that are popular in modern engineering. The ESO approach has also been modified for boundary based optimisation, where the design variables are the location of spline control points and analysis is performed by the boundary element method (Cervera & Trevelyan 2005). A few other methods also use spline control points as design variables in structural topology optimisation (S. Lee et al. 2007; S. Lee & Kwak 2008; Edwards 2008). However, these methods require careful handling to merge and split splines when merging or splitting holes to alter the topology. Furthermore, the boundary can move significantly during optimisation, which can lead to a bunching or spreading of control points and a poor boundary representation. Therefore, it appears that current spline based methods are not suitable for structural topology optimisation.

The main alternative to the spline based approach is to use an implicit function to describe the boundary. The implicit function is usually discretized at the nodes of a grid and the boundary is often defined by points within the grid where the

implicit function is zero. This method of boundary representation is particularly useful for topology optimisation as it naturally allows for merging and splitting of holes. Methods using implicit boundary representation are often referred to as level set methods, because the boundary can be defined as the zero level set of a higher dimensional implicit function. The level set method for structural topology optimisation has received increasing interest over the last decade and a detailed review is presented in Section 2.3.

2.2.4 Conclusions

Topology optimisation is the most general approach to structural optimisation, allowing for the layout, shape and some sizing of the structure to be considered during optimisation. The development of structural topology optimisation began with truss type structures. However, solutions are limited by the nodal locations and the complexity of the ground structure. Therefore, topology optimisation was developed for continuum type structures, which started with element based methods, such as homogenisation and SIMP. These methods have been successfully applied to solve numerous structural optimisation problems, although they can suffer from numerical problems, such as checkerboard pattern formation, and solutions often possess unclear "fuzzy" boundaries that can be difficult to interpret.

The limitations of element based methods has led to research in boundary based topology optimisation. Boundary based methods inherently produce solutions with a clearly defined boundary and do not appear to suffer form checkerboard pattern formation. Boundary based methods are sometimes extensions of shape optimisation techniques to include some facility to alter topology. However, these approaches usually use splines to represent the boundary, which presents difficulties when the topology is altered by merging or splitting of holes. Also, large movement during optimisation can lead to bunching

or spreading of control points that can result in poor boundary representation. The alternative implicit boundary representation can handle topology changes more naturally. Therefore, recent interest in the boundary based approach to topology optimisation has focussed on implicit boundary representation, particularly the level set method, which is discussed in more detail in the following section.

2.3 Level set based structural optimisation

The level set method was originally developed as a mathematical tool for computing the motion of interfaces in two or three dimensions (Sethian 1999; Osher & Fedkiw 2003). The method represents interfaces using an implicit function that can be easily updated under a velocity field to track the motion of the interfaces. This approach naturally allows for complicated phenomena to occur, such as interface merging or splitting. The simple, but flexible qualities of the level set method are attractive to a wide range of applications where the dynamic movement of interfaces has to be accurately and efficiently computed.

The level set method is also well suited to boundary based topology optimisation, where interfaces becomes structural boundaries and the velocity field is derived from shape sensitivity analysis to propagate the design towards an optimal design. This section presents fundamental theoretical aspects of the level set method in the context of structural optimisation. The various approaches to level set optimisation are then critically reviewed.

2.3.1 Fundamental principles

The first ingredient common to all level set based methods is to define the structure using an implicit function, $\phi(x)$:

$$\begin{cases} \phi(x) \geq 0, & x \in \Omega_S \\ \phi(x) = 0, & x \in \Gamma_S \\ \phi(x) < 0, & x \notin \Omega_S \end{cases} \tag{2.1}$$

where Ω_S is the domain of the structure and Γ_S the boundary of the structure. The implicit function is usually discretized at the nodes of a grid and appropriately interpolated between them. The initial function is often defined as the signed distance between the grid point and the boundary. The implicit function can be updated under a velocity field over time by iteratively solving a Hamilton-Jacobi type equation:

$$\frac{\partial \phi(x,t)}{\partial t} + \nabla \phi(x,t) \frac{dx}{dt} = 0 \tag{2.2}$$

This equation can be discretized and rearranged to produce an update formula that can be employed to progressively optimise a structure using an iterative approach:

$$\phi_i^{k+1} = \phi_i^k - \Delta t \, |\nabla \phi_i^k| V_{n,i} \tag{2.3}$$

where k is the current iteration, Δt is a discrete time step, i is a node where the discrete implicit function is defined and $V_{n,i}$ is the component of the discrete velocity field, or function, normal to the boundary. Note that the comma in the subscript does not indicate differentiation. The sign convention, used throughout this thesis, is that a positive velocity value acts inwards from the boundary. It is well known that the change in shape through a velocity function is only dependent on the component normal to the boundary. Thus only this part needs to be considered during optimisation (Sethian 1999; Allaire et al. 2004).

Shape sensitivity analysis

The velocity function is usually derived from shape sensitivity analysis so that the updated structure is improved with respect to the optimisation objective and constraints. Shape sensitivity analysis theory is reasonably involved, thus, only theoretical aspects and results relevant to the work in this thesis are discussed.

This thesis is only concerned with linear elastic structures. The behaviour of a linear elastic structure with prescribed loading and boundary conditions can be determined by solving the following virtual work balance equation:

$$\begin{aligned} &\int_{\Omega_S} \tilde{E_{ijkl}}\, \varepsilon(u)_{ij}\, \varepsilon(v)_{kl}\, d\,\Omega_S = \int_{\Omega_s} b\, v\, d\ \Omega_S + \int_{\Gamma_s} f\, v\, d\,\Gamma_S \\ &u|_{\Gamma_D} = u_0 \quad \forall v \in U \end{aligned} \tag{2.4}$$

where $\tilde{E}_{ijkl}$ is the material property tensor, $\varepsilon(u)_{ij}$ the strain tensor under displacement field u in the space U of kinematically permissible displacement fields subject to prescribed displacements u_o along the boundary, b are body forces and f are surface tractions. Structural problems often only consider zero displacement boundary conditions, $u_o = 0$, thus only this type of condition is considered here. For practical problems (2.4) can be solved computationally using the FEM. If the structure domain and boundary are defined using the implicit function, (2.4) becomes:

$$\begin{aligned} &\int_{\Omega} \tilde{E_{ijkl}}\, \varepsilon(u)_{ij}\, \varepsilon(v)_{kl}\, H(\phi)\, d\,\Omega = \int_{\Omega} b\, v\, H(\phi)\, d\,\Omega + \int_{\Gamma_s} f\, v\, d\,\Gamma_S \\ &u|_{\Gamma_D} = 0 \quad \forall v \in U \end{aligned} \tag{2.5}$$

where Ω is the domain containing all possible solutions, such that $\Omega_S \subset \Omega$, Γ_D is the part of boundary subject to zero displacement boundary conditions and $H(\phi)$ is the Heaviside step function:

$$H(\phi) = \begin{cases} 1, & \phi \geq 0 \\ 0, & \phi < 0 \end{cases} \tag{2.6}$$

An arbitrary objective function is considered, with the form:

$$F(u, \phi) = \int_{\Omega} J(u)\, H(\phi)\, d\,\Omega + \int_{\Gamma_s} L(u)\, \Gamma_S \tag{2.7}$$

where the functions $L(u)$ and $J(u)$ are assumed differentiable with respect to the geometry or shape of the structure domain, Ω_S, if the domain is smooth and bounded. The result of the differentiation is a shape derivative, which provides information about the change in objective function with respect to a change in the

shape or geometry of the structure domain. The classic shape derivative for the arbitrary objective function is (Allaire et al. 2004):

$$\begin{aligned} F'(u,\phi) = & \int_{\Gamma_N} \left(\tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(p)_{kl} - \frac{\partial(f\,p)}{\partial n} - \bar{\kappa}(f\,p) - p\,b - J(u) \right) V_n \, d\Gamma_N \\ & - \int_{\Gamma_S} \left(\frac{\partial L(u)}{\partial n} + \bar{\kappa}\,L(u) \right) V_n \, d\Gamma_S - \int_{\Gamma_D} \left(\tilde{E}\varepsilon(u)\varepsilon(p) + J(u) \right) V_n \, d\Gamma_D \end{aligned} \tag{2.8}$$

where $\Gamma_S = \Gamma_N \cup \Gamma_D$, $\bar{\kappa}$ is the mean curvature of the boundary, n a unit normal vector to the boundary and p is the solution to the adjoint equation:

$$\begin{aligned} \int_\Omega \tilde{E}_{ijkl}\varepsilon(p)_{ij}\varepsilon(v)_{kl} H(\phi)\, d\Omega = & \int_{\Gamma_S} \frac{\partial L(u)}{\partial u} v \, d\Gamma_S \\ & + \int_\Omega \frac{\partial J(u)}{\partial u} v \, H(\phi)\, d\Omega \quad , \quad p|_{\Gamma_D} = 0 \quad \forall v \in U \end{aligned} \tag{2.9}$$

For practical reasons the portion of the boundary subject to surface tractions and displacement boundary conditions is often fixed during optimisation. Therefore, the Γ_N part of boundary is split so that $\Gamma_N = \Gamma_F \cup \Gamma_0$, where Γ_F is the part subject to surface tractions and Γ_0 the free part of the boundary that is permitted to move during optimisation. To fix Γ_F and Γ_D during optimisation the velocity function is defined to be zero along those parts of the boundary. Under this condition, the shape derivative (2.8) simplifies to:

$$F'(u,\phi) = \int_{\Gamma_0} \left(\tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(p)_{kl} - \frac{\partial L(u)}{\partial n} - \bar{\kappa}\,L(u) - pb - J(u) \right) V_n \, d\Gamma_0 \tag{2.10}$$

Shape sensitivity for the compliance function

The compliance objective function (1.1) can be rewritten in an equivalent form using the solution to the static equilibrium constraint:

$$\begin{aligned} C(u,\phi) = & \int_\Omega \tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(u)_{kl} H(\phi)\, d\Omega \\ = & \int_\Omega b\,u\,H(\phi)\, d\Omega + \int_{\Gamma_S} f\,u\, d\Gamma_S \end{aligned} \tag{2.11}$$

Comparing function (2.11) to the arbitrary objective function (2.7), $L(u) = f\,u$ and $J(u) = b\,u$. Therefore, by substituting these functions into (2.9) and comparing to (2.5) it is evident that $p = u$ and the problem is self adjoint. This is a well known result and (2.10) can be used to compute the shape derivative of the compliance problem (Allaire et al. 2004). Noticing that $L(u) = 0$ on the free part of the boundary, the shape derivative of the compliance function becomes:

$$C'(u,\phi) = \int_{\Gamma_0} \left(\tilde{E_{ijkl}} \varepsilon(u)_{ij} \varepsilon(u)_{kl} - 2\,b\,u \right) V_n \, d\,\Gamma_0 \tag{2.12}$$

The shape sensitivity, ς of the compliance function along the free boundary is defined here as:

$$\varsigma(u) = \tilde{E_{ijkl}} \varepsilon(u)_{ij} \varepsilon(u)_{kl} - 2\,b\,u \tag{2.13}$$

The goal of the optimisation problem is to minimise the compliance function (1.1). Thus, the velocity function can be simply defined from the shape sensitivity to produce a negative sign of the shape derivative (2.12):

$$V_n = -\varsigma(u) = 2\,b\,u - \tilde{E_{ijkl}} \varepsilon(u)_{ij} \varepsilon(u)_{kl} \tag{2.14}$$

The velocity function is then used to update the implicit function using (2.3), thus improving the structure with respect to the objective. However, the velocity function does not account for the problem constraint. The most common approach to handle constraints is to transform the constrained problem into an unconstrained one using the Lagrange multiplier method. The unconstrained compliance problem is then:

$$\text{Minimise : } \bar{C}(u,\phi) = C(u,\phi) + \lambda \int_{\Omega} H(\phi)\, d\,\Omega \tag{2.15}$$

where λ is a positive Lagrange multiplier. The shape derivative of the unconstrained problem can be easily evaluated using (2.12) and the shape derivative for the volume function (Allaire et al. 2004):

$$\bar{C}'(u,\phi)=\int_{\Gamma_0}\left(\varsigma(u)-\lambda\right)V_n\,d\,\Gamma_0 \tag{2.16}$$

Therefore the velocity function can be simply redefined as:

$$V_n=\lambda-\varsigma(u) \tag{2.17}$$

A fixed value of λ can be used (Allaire et al. 2004), although, this does not necessarily guarantee constraint satisfaction. An alternative is to compute λ during each iteration, assuming the volume of the structure is conserved during boundary propagation (M.Y. Wang et al. 2003). However, this approach can encounter difficulties, as preserving volume or mass using the level set method can be problematic (S.Y. Wang et al. 2007). The approach can be improved by heuristically penalising the velocity function depending on the state of the volume constraint (S. Y. Wang & M. Y. Wang 2006a). A more robust approach is to use a bi-sectioning algorithm to compute a λ value that exactly satisfies the constraint each iteration (S.Y. Wang et al. 2007).

Numerical aspects

Efficient and robust numerical techniques have been developed to evaluate the discretized level set update equation (2.3). Firstly, the time step, Δt in (2.3) should be limited by well known Courant–Friedrichs–Lewy (CFL) condition for stability:

$$\Delta t=\beta h/|V_{n,i}|_{max} \tag{2.18}$$

where $0<\beta<1$ and h is the discrete grid spacing.

The velocity function derived from shape sensitivities (2.17) is only applicable to the points along the structural boundary. In order to update the level set function using (2.3), discrete velocity values, $V_{n,i}$ have to be defined at all grid points using a velocity extension scheme. The simplest scheme is to compute sensitivities everywhere in the domain, thus naturally defining the velocity function (Allaire et al. 2004; S. Y. Wang & M. Y. Wang 2006b; Z. Luo et al. 2007). This natural extension velocity approach requires the strain field to be

available outside the structure domain. This can be achieved by filling the void region of the design domain with a weak material (Allaire et al. 2004). Alternatively, the strain field can be smoothed over the structure boundary allowing sensitivities to be computed in a region just outside the structure (S. Y. Wang & M. Y. Wang 2006a). However, the implicit function can easily stray from being a signed distance function when using the natural extension scheme. This can lead to an implicit function that is too steep or shallow, thus reducing the accuracy and stability of the level set method (Allaire et al. 2004). Therefore, the implicit function must be frequently re-initialised to a signed distance function to maintain stability.

To avoid frequent re-initialisation, an extension velocity scheme that maintains the signed distance function can be employed (Adalsteinsson & Sethian 1999). This method extrapolates velocities out from the boundary using an efficient fast marching scheme. However, to avoid evaluating the velocity function everywhere in the domain for each iteration, the extension computation can be limited to a narrow band around the boundary (Sethian 1999). This makes the velocity computation proportional to the boundary length rather than the area of the design domain (M.Y. Wang et al. 2003). However, the signed distance property is now only maintained within the narrow band region, thus re-initialisation is still required when the boundary approaches the edge of the narrow band region.

The gradient of the discretized implicit function is usually computed numerically using a finite difference scheme. Level set methods often employ an upwind finite difference scheme that estimates the gradient based on the direction of movement of the implicit function (Osher & Fedkiw 2003). This promotes stability during the propagation of the boundary, as information travels in the direction of the boundary movement.

Finite element analysis (FEA) is usually employed to compute the primary and adjoint states of the problem used for sensitivity computation (2.10). Traditional

FEA methods fit elements to the geometry of the structure, for example (Cook et al. 2002). However, this fitted mesh approach is often not suitable for topology optimisation, as the entire structure would have to be frequently re-meshed, which is inefficient and not straightforward (S.Y. Wang & M.Y. Wang 2006a). Thus, level set optimisation methods often adopt a fixed mesh FEA approach, where the design domain is discretised by regular elements that remain fixed throughout optimisation (Allaire et al. 2004; M.Y. Wang et al. 2003). For simplicity and convenience the fixed mesh used for analysis often coincides with the grid used to discretise the implicit function. However, the structure geometry is not constrained to conform to the fixed mesh, thus discontinuous elements can exist at the boundary interface. This poses a challenge to accurate analysis along the boundary and is discussed in the next section.

2.3.2 Review of level set optimisation approaches

The first approach that utilised the level set method for structural design was proposed by Sethian & Wiegmann (2000). The aim of this method was to improve, rather than fully optimise the design of the structure. The approach adopted the fully stressed design methodology, where the aim is to achieve a uniform stress over the entire structure. This criterion was used to heuristically define the velocity function along the boundary and also allowed for hole creation by simply removing under used material within the structure. Another early method was aimed at optimising the design of a two density drum for frequency response (Osher & Santosa 2001). This approach used a more analytic velocity function defined from functional gradients of the objective.

A significant advancement of level set based structural optimisation was to employ shape sensitivity analysis to define the velocity function (Allaire et al. 2004; M.Y. Wang et al. 2003), as shown in Section 2.3.1. This concept was developed independently by two groups whose approaches were reasonably

similar. These methods are a direct application of the original level set method and use numerical procedures developed for general applications (Sethian 1999; Osher & Fedkiw 2003).

The method proposed by Allaire et al. (2004) suggested performing one shape sensitivity computation, then updating the level set function several times until the objective function ceased decreasing, then a new sensitivity computation was performed. This approach was motivated by the observation that computing the update is cheap compared to computing sensitivities. However, the evaluation of an objective for a structural optimisation problem usually requires assembling and inverting the stiffness matrix each time the structure is updated. The FEA process is often the bottleneck in topology optimisation methods (X. Wang et al. 2004; J. Luo et al. 2008; Edwards 2008), thus this approach does not seem to allow for significant efficiency gains. Furthermore, once the FEA has been performed it seems reasonable to continue and compute sensitivities, which may increase accuracy due to more frequent sensitivity analysis. Indeed this approach was adopted by the contemporary method (M.Y. Wang et al. 2003).

The direct approach that combines the level set paradigm with shape sensitivity analysis produces good solutions to benchmark problems, with smooth boundaries and no checkerboard patterns. However, the direct approach has some limitations and drawbacks in the context of structural topology optimisation. The three most significant limitations are: slow convergence, lack of new hole creation facility and inaccurate analysis when using the fixed grid approach.

Slow convergence of the level set method to an optimum is partly because the method was originally developed to dynamically track fronts where the position of the boundary each iteration may be important. However, for optimisation, intermediate solutions are of no real value, as we are only interested in final solution. Furthermore, the conventional approach to solve the level set equation (2.2) is constrained by the CFL condition that limits the movement of the

boundary each iteration by resolution of the mesh (2.18). This effect becomes worse if a finer mesh is used to increase analysis accuracy.

The second limitation of the direct level set method is that new holes cannot be created within the structure to alter the topology, at least for two dimensional problems (Allaire et al. 2004; Mei & X. Wang 2004). Topology can only change by merging or eliminating existing holes. This leads to solutions being dependent on the topology of the initial design.

Lastly, the use of the simple and efficient fixed mesh approach to perform FEA leads to approximated elements along the boundary. This suggests that errors in the analysis will be greatest at the boundary where approximated elements are present. However, the optimisation process is driven by sensitivities computed from FEA along the boundary. Therefore, accurate boundary sensitivity computation using a fixed mesh is a challenge for the level set method.

The three limitations of the direct approach identified above have been addressed in subsequent literature by employing a variety of ideas and techniques. The following review concentrates on each limitation in turn, describing and comparing the various approaches employed to overcome them.

Approaches to improve efficiency

A simple approach to increase the efficiency of the direct method is to modify the velocity function such that a greater movement of the boundary occurs during each iteration. This amounts to using a non-linear velocity map that is non-decreasing and preserves the sign of the original velocity function (X. Wang et al. 2004). This simple approach can speed up convergence by 2-3 times. For example, the following velocity function was proposed by Rong & Liang (2008) using the non-linear map approach:

$$\overline{V}_n = \lambda - \varsigma \exp\left(1 - |\varsigma / \varsigma_m|\right) \tag{2.19}$$

where $\overline{V}_n$ is the modified velocity function and ς_m is the maximum sensitivity value.

Beyond the non-linear map approach, two alternative methods have been proposed to solve the Hamilton-Jacobi type level set equation (2.2) that alleviate the CFL condition and allow for larger time steps. The Semi-Lagrangian approach solves the level set equation (2.2) by first approximating the upstream point of the implicit function using the velocity function and a set time step (Xia et al. 2006). The upstream point of the implicit function is then interpolated to find the updated value. This approach allows for larger time steps compared to the conventional solution method, which can led to a significant reduction in convergence time. However, oscillations can occur during optimisation and an adaptive time step may be required to decreases boundary movement near the optimum.

The Additive Operator Scheme (AOS) works by splitting the spatial operator into one dimensional components in orthogonal directions (J. Luo et al. 2008). The solution to the multidimensional level set equation is then the average of the individual one dimensional solutions. This scheme is a semi-implicit method for solving the level set equation that is stable for all time steps and is thus not constrained by the CFL condition. However, experience is required to choose an appropriate time step. If too large a time step is chosen then oscillation can occur in the optimisation process, especially near the optimum, and the topological complexity of the solution may also be affected. However, an appropriate choice of time step can result in up to a ten fold increase in efficiency compared with the conventional solution method.

The above approaches can be described as modifications of the direct level set method, as most of the general framework remains unaltered. However, there are a number of level set based structural optimisation methods that diverge significantly from the direct approach, which may also improve efficiency. The

spectral level set method (Gomes & Suleman 2006; Gomes & Suleman 2008) uses coefficients of a Fourier series of the implicit function, $\phi(x)$ as design variables for the optimisation problem. This effectively reduces the design space to small number of variables compared to the entire discretized implicit function. However, the proposed solution method does not utilise sensitivity data, instead employing a random search algorithm. This approach can require a large number of analyses to find the optimum solution, which is inefficient compared to other methods.

Some methods use topological derivatives to find the optimum design, without considering shape sensitivities (Amstutz & Andra 2006; Norato et al. 2007). Topological derivatives describe the sensitivity of the objective function when a small hole is inserted into the structure. This approach facilitates the update of the entire design during each iteration, allowing for more dramatic changes compared to an update based purely on boundary movement. This approach can also be used to insert structural material into the void part of the design space, adding flexibility to the method. However, this approach lacks an efficient and robust method for including constraints beyond a simple volume constraint.

Radial basis functions (RBFs) have been used to interpolate the discretized implicit function (S. Y. Wang & M. Y. Wang 2006b; S. Y Wang et al. 2007). The optimisation problem can then be formulated and solved using efficient mathematical programming methods, where the design variables are the expansion coefficients of the RBFs. For example the optimality criteria method can be employed (Z. Luo et al. 2008b), although this requires appropriate choice of a damping factor and move limit on design variables to stabilise the optimisation process. Alternatively, the method of moving asymptotes can be employed, which is also often used in element based methods (Bendsøe & Sigmund 2004). The multi-quadratic spline RBF was initially used to perform the interpolation. However, this leads to inverting a dense co-location matrix each iteration and the solution can be effected by the choice of the free shape

parameter. The RBF approach was improved by employing compactly supported RBFs that produce a more sparse co-location matrix and do not require a free shape parameter (Z. Luo et al. 2007).

Approaches for hole insertion

A popular mechanism to facilitate hole insertion is to use topological derivatives to identify favourable positions for inserting new holes when using level set based optimisation. However, linking topological derivatives to the direct level set method is not trivial and various approaches have been proposed. Topological derivatives can be added to the level set equation (2.2) using a forcing term that decreases the implicit function inside the structure for negative derivative values and increases it for positive values (Burger et al. 2004; He et al. 2007; Challis 2010). This allows holes to emerge over a number of iterations in favourable locations. The forcing term approach can be extended to allow structural material to emerge in the void region of the design domain. However, the strength of the forcing term can affect the efficiency and stability of the method. This problem can be avoided by introducing a switching mechanism, such that topological derivatives are only considered if they provide sufficient improvement in the objective. A further problem arises if the implicit function is initialised as a signed distance function, as this makes it more difficult for holes to emerge further from the boundary. Therefore, different parts of the structure can be erroneously treated unequally with respect to hole creation.

Topological derivatives can also be used to exchange material between regions at a set percentage to allow for hole insertion and reinforcement of the void region during optimisation (Rong & Liang 2008), although the choice of exchange percentage is heuristic. Another approach is to use topological derivatives during each iteration to create new holes if the volume constraint is violated (Zhuang et al. 2007; Myslinski 2008). This can be inefficient when small holes are created every iteration, which are then closed during subsequent boundary movement.

Therefore, in practice hole insertion using topological derivatives is usually only considered after an arbitrary set number of boundary updates have been completed (Allaire et al. 2005; Takezawa et al. 2010). However, the number of boundary updates between hole insertions can affect the solution obtained and efficiency of the method. Furthermore, for the compliance objective, the topological derivative indicates that the creation of any hole increases the objective. However, subsequent shape optimisation of a new hole can reduce the objective overall (M. Kim et al. 2009). Thus, it is unclear if a new hole will become a feature in the optimal design, or simply disappear due to boundary optimisation.

Exclusive use of topological derivatives to solve structural optimisation problems has also been proposed (Amstutz & Andra 2006; Norato et al. 2007), without considering shape derivatives. The advantage of this approach is that optimality criteria methods can be employed to solve the optimisation problem. Although including constraints other than the simple volume constraint can be difficult. Topological derivatives offer a useful mechanism to determine the optimum locations for the insertion of new holes. However, the link between shape and topological derivatives in level set based optimisation is unclear, as existing methods either employ some heuristic parameter, such as a forcing term or number of boundary updates between hole insertions, or simply abandon shape derivatives.

Heuristic schemes based on removing under-used material have also been employed to facilitate hole insertion in level set based optimisation. An evolutionary approach was employed to remove areas of low stress at a removal rate that was dynamically updated during the optimisation (Sethian & Wiegmann 2000). This allowed new holes to be created in locations where stresses were low, indicating under-used material. The adaptive inner front level set method is a similar approach, where new inner fronts, or holes, were created by removing material below a level set of the strain energy density (K. Park & Youn 2008). The amount of material removed was controlled by setting a target volume to be

obtained after hole insertion. A genetic algorithm has also been used to randomly mutate the topology by adding and removing elements (Rong & Liang 2008). Thus, holes can emerge if internal elements are removed from the structure.

The above approaches are designed to allow new hole creation for the direct level set method, except for the exclusive use of topological derivatives. However, some of the alternative level set approaches discussed under efficiency improvement serendipitously have an inherent mechanism that allow new holes to be created during optimisation. The spectral level set method does not rely on shape sensitivities or boundary propagation to optimise the structure (Gomes & Suleman 2006). Instead the design variables are coefficients of a finite Fourier series, which can be changed to alter the topology and introduce new holes. Although the topology is restricted by the number of design variables.

The RBF level set method can create new holes if the natural velocity extension method is employed. This allows holes to emerge anywhere in the structure throughout the optimisation (S. Y. Wang & M. Y. Wang 2006b; Z. Luo et al. 2008a). However, this requires the computation of sensitivities over the entire structure during each iteration. Alternatively, the shape sensitivity can be computed using a volume integral approach in a narrow region around the boundary (Z. Luo & Tong 2008; Z. Luo et al. 2009a). This allows holes to emerge in the boundary region, which can then migrate further into the structure in subsequent iterations.

The AOS method for solving the level set equation (2.2) allows the boundary to merge and split, producing new holes near the boundary during optimisation. The AOS scheme has also been employed in a level set method based on phase field methodology, where the design is composed of two distinct phases, solid and void (Wei & M.Y. Wang 2009; Z. Luo et al. 2009b). With this approach new holes can easily appear within the design, as the two phases are free to exchange anywhere within the design domain. This has the additional benefit of inserting

solid phase material into void regions. However, solutions obtained using AOS methods are dependent on the strength of the diffusion or regularization term used and may still be partially dependent on the initial design.

As discussed above, a number of techniques have been implemented that allow hole insertion in level set based topology optimisation methods. However, the link between shape and topological derivatives is unclear and the optimum topology of alternative methods can be sensitive to a choice of parameters.

Approaches for boundary sensitivity computation

Level set methods often employ a fixed mesh for simplicity and efficiency when performing FEA and sensitivity computations. However, this introduces intersected elements along the structure boundary. A common approach to handle intersected elements is to simply weight their stiffness by the proportion of structural material within the element. This is often called the ersatz material approach (Allaire et al. 2004; S. Y Wang et al. 2007; J. Luo et al. 2008), but can also be described as the density or area weighted method. This method is simple, but cannot capture the exact geometry of the boundary in the analysis and can have a destabilising effect on optimisation due to poor computation of boundary sensitivities (Jang et al. 2004; S. Y. Wang & M. Y. Wang 2006a; Z. Liu et al. 2005; Wei et al. 2010). It is therefore common to smooth or regularise the velocity function to avoid numerical problems and improve reliability of convergence. For example the velocity function can be smoothed across the boundary discontinuity using a simple linear filter (S.Y. Wang et al. 2007) or it can be regularised by including a term dependent on mean curvature (J. Luo et al. 2008). Also, the maximum strain energy sensitivity value can be capped throughout the optimisation to help smooth the sensitivity and velocity distribution (Z. Liu et al. 2005). However, these smoothing techniques do not address the fundamental problem of poor sensitivity computation by the area weighted method and introduce additional numerical parameters that can influence optimum solutions.

The selection of these parameters is often problem dependent and can be difficult.

Another common approach to fixed mesh FEA is to smooth the discontinuity at the boundary interface using a smoothed approximate Heaviside function (M.Y. Wang et al. 2003; Belytschko et al. 2003b). This function is then employed when numerically integrating the stiffness matrix for elements near the boundary. However, this approach produces less accurate stiffness matrices for those elements near the boundary, which can lead to poor sensitivity computation.

The geometry projection method represents the boundary discontinuity using a geometry measure based on a smoothed variation of local element densities (Norato et al. 2004; Norato et al. 2007). If the filter window is equal to the element edge length then this approach is analogous to the density weighted method, although the boundary geometry is more accurately accounted for by the geometry measure. This main advantage of this approach is the convergence to the real problem in the limit of mesh refinement.

The eXtended FEM (X-FEM) augments shape functions near discontinuities, such as cracks, inclusions and material interfaces (Belytschko et al. 2001; Belytschko et al. 2003a). This approach has been employed in level set methods to model the boundary interface within intersected elements and has been found to outperform the density weighted method (Van Miegroet & Duysinx 2007; Wei et al. 2010). However, accurate numerical computation of the stiffness matrix requires a quadrature rule that accounts for the discontinuity, such as dividing the element into polygons that match the interface. Although a less accurate computation can be performed using a simpler numerical integration scheme that does not directly account for the interface (Belytschko et al. 2001).

Another approach, similar to X-FEM, is to fit shape functions to the boundary interface within intersected elements (Jang et al. 2004; Jang & Y. Y. Kim 2005). However, for two dimensional structures, non-quadrilateral internal areas require special treatment using degenerate quadrilateral element formulations to compute

the stiffness matrices. A multi-scale method is also employed to improve mesh resolution and accuracy at the boundary interface.

The above approaches for performing fixed mesh FEA and computing sensitivities all rely on modifying the stiffness matrix computation for elements intersected by the boundary. Thus, the number of elements, degrees of freedom and geometry of the mesh remain constant throughout optimisation. There are a number of alternative approaches that go beyond this restriction to improve the accuracy of the analysis at the boundary interface. The superimposed FEM uses a local mesh of triangular elements to further discretize the fixed mesh elements intersected by the boundary (S. Y. Wang & M. Y. Wang 2006a). The local mesh is then coupled with the global fixed mesh during integration using a double mapping scheme. However, the velocity function derived from sensitivities is still smoothed over the boundary discontinuity using a simple linear filter.

Boundary elements have been employed to discretize the boundary and perform analysis in level set based optimisation (Abe et al. 2007). The main advantage of the boundary element method, compared to the FEM, is the reduction in degrees of freedom in the discretization. In the approach, linear elements were placed along the boundary at regular intervals, determined by the ratio of total boundary length to the width of an element in the fixed background mesh. The fixed mesh was used to discretize the implicit function and perform the level set update using velocities extended from the boundary element analysis. However, the arrangement of boundary elements has to be re-defined each iteration as the boundary moves and convergence was very slow for the examples presented.

Ha & Cho (2008) used an unstructured fitted mesh to analyse and optimise geometrically non-linear structures with the level set method. A mesh that fitted the exact boundary geometry was automatically constructed each iteration using triangular elements and Delaunay triangulation. The velocity function computed

from the fitted mesh was then extended to an underlying fixed regular grid using a distance weighted interpolation method. The extended velocities on the fixed grid were then used to update the implicit function, which was discretized on the fixed grid as usual. However, constructing the fitted mesh during each iteration is computationally expensive and only suitable for non-linear structures.

An adaptive moving mesh method was implemented to improve the mesh resolution at the boundary interface (Z. Liu & Korvink 2008). This approach avoids specific re-meshing during each iteration as the mesh topology (number and connectivity of nodes) is maintained. The mesh is simply altered in each iteration by moving nodes throughout the design domain to obtain the best approximation of the global solution. A fixed mesh with the same topology as the moving mesh is used to discretize the implicit function and perform the level set update. Thus, the relation between the moving and fixed meshes can be mapped using a Jacobian matrix. This simplifies the translation of quantities between the two meshes, such as the implicit function. However, this approach was not found to be efficient when compared to a dense fixed mesh, especially if the loading conditions were also fixed. The approach is more suitable if loads are design dependent, such as pressure or self-weight, as the mesh moves to provide a better resolution at the boundary.

The range of methods employed to perform analysis for level set based structural optimisation can be categorised as either those that modify the stiffness matrix formulation of intersected elements in a fixed mesh, or those that re-mesh or move the mesh to provide a more accurate discretisation near the boundary. The various approaches also provide a trade-off between accuracy and efficiency. This ranges from the simple area weighted approach, which is very efficient, but inherently inaccurate at the boundary, to the unstructured fitted mesh approach, which can provide a more accurate analysis, but constant re-meshing makes it inefficient. For structural optimisation the FEA process can be a bottleneck, often taking much longer to perform than other components, such as updating (X. Wang

et al. 2004; J. Luo et al. 2008; Edwards 2008). Therefore, the most efficient analysis method that provides sufficient accuracy should be employed to help yield an efficient optimisation method.

2.3.3 Conclusions

The idea of using shape sensitivity analysis to compute velocities for direct use of the level set method for structural optimisation provides a method inherently able to handle significant topology changes and produces solutions with clear boundaries. However, the direct approach has three main limitations. It can be inefficient and slow to converge, solutions can be dependent on the initial design, due to lack of new hole creation, and performing accurate analysis along the boundary using a fixed mesh is a challenge. There has been a significant amount of literature addressing these three points and a number of different methods and approaches have been proposed, ranging from simple modification of the direct approach to different paradigms. However, there does not appear to be any agreement on one particular approach, although perhaps this may emerge as research continues. Despite the proposal of different level set optimisation paradigms, the direct approach is still attractive, as there are robust and efficient numerical procedures readily available that have been developed for other applications of the level set method.

2.4 Uncertainty in topology optimisation

Including uncertainty in structural design and optimisation is important to produce efficient structures that are robust and reliable when operating in real world environments. Various uncertainties can affect the reliability and robustness of a structure, including loading conditions, material properties and geometry (S. Choi et al. 2007). The safety factor is the traditional engineering approach to account for uncertainty. However, this heuristic approach does take direct account of uncertainty and can lead to an overly conservative or under designed structure. Therefore, there is increasing interest in probabilistic methods to take a more direct account of uncertainty. Various probabilistic approaches have been developed to account for different types of uncertainty in structural design and optimisation methods (S. Choi et al. 2007; G-J. Park et al. 2006), however, these have not been applied to structural topology optimisation until recently.

There are two main approaches for including uncertainty that have been applied to structural topology optimisation. The first is to introduce a number of reliability constraints based on probability of failure, often referred to as Reliability-Based Design Optimisation (RBDO). The second approach is robust optimisation, where a probabilistic objective function is considered, such as the expected or worse case performance. There are also various types of uncertainty that can affect structural performance including loading, geometry and material properties (S. Choi et al. 2007). The purpose of the following sections is to review how RBDO and robust optimisation have been applied to structural topology optimisation, including the various types of problems and uncertainties that have been considered.

2.4.1 Reliability-based topology optimisation

The first applications of the reliability paradigm in structural topology optimisation were a direct use of RBDO methods. Objective functions were to minimise structural weight whilst satisfying a constraint that the probability of failure is less than a prescribed small amount. Failure was defined by negative values of a limit state function. For example a stress based limit state function, $G(x)$ could be defined as:

$$G(x)=\sigma_{max}-\sigma(x) \tag{2.20}$$

where σ_{max} is the upper limit on a critical stress value and x is a vector of the uncertain variables. The failure probability constraint is usually recast to ensure the reliability of the structure is greater than a required value. This is usually achieved using a reliability index, which is a measure on the number of standard deviations a failure point is from the mean. For a problem with many variables the failure points of the limit state function form a hyper-surface, or failure surface. This can make the computation of the reliability index difficult. However, the computation can be simplified using an approximation of the limit state function. Assuming the variables, x are statistically independent, the limit state function can be approximated by a first order Taylor expansion at the mean, μ_x, which is exact if the function is linear:

$$G(x)\approx G(\mu_x)+\nabla G(\mu_x)^T(x-\mu_x) \tag{2.21}$$

This is the First Order Reliability Method (FORM) for computing an approximation to the failure point and reliability index. The main benefit of FORM is the availability of analytical sensitivities for the reliability constraint. This approach leads to a straightforward algorithm for solving RBDO problems, involving an inner loop, where the critical failure point is found by minimising the distance between the mean and the limit state function surface. For stability the

inner loop is usually performed using uncertain variables that are normalised such that the mean is zero. This is the Reliability Index Approach (RIA), where the constraint is for the reliability index to be greater than a prescribed value.

The RIA is straightforward, but involves differentiation of the reliability index with respect to the design variables, which can be complex (C. Kim et al. 2006), and a solution to the inner loop problem is not guaranteed (Maute & Frangopol 2003). Therefore, it is usually replaced with the equivalent Performance Measure Approach (PMA), which places a constraint on the limit state function being positive when the prescribed reliability index is exactly satisfied. The PMA gives similar solutions to those found using the RIA, but sensitivity analysis is simpler and convergence more stable. The FORM approach has been used in topology optimisation for reliability constraints on stiffness, fundamental eigenfrequency and critical displacements (Maute & Frangopol 2003; Jung & Cho 2004; Mogami et al. 2006; C. Kim et al. 2006). When compared to a Monte Carlo simulation FORM was found to be accurate for stiffness reliability constraints, but not necessarily for eigenvalues, although FORM did provide an improved design compared to the deterministic solution (Mogami et al. 2006).

The efficiency of Reliability-Based Topology Optimisation (RBTO) has been improved by employing a single loop approach that removes the inner loop computation of the critical failure point each iteration (Silva et al. 2010; Kang & Y. Luo 2010). The basis of the single loop approach is to update the normalised uncertain variables used in finding the critical failure point simultaneously with design variables each iteration. Therefore, the reliability constraint and solution to the inner loop problem do not have to be exactly satisfied during optimisation, but only for the final solution. This can reduce computational cost to be similar to a deterministic problem (Silva et al. 2010). The RBTO method has also been generalised to include non-probabilistic uncertainty models (Y. Luo et al. 2009; Kang & Y. Luo 2009; Kang & Y. Luo 2010). The approach is to assume that uncertain variables can be defined using convex models when probability data is

unavailable.

There are some alternative approaches to the RBTO method outlined above. The reliability constraint can be used to simply change the uncertain parameters when solving a deterministic problem. Parameters are changed based on their sensitivity with respect to the objective function and the value of the reliability target (Kharmanda et al. 2004; Zhang & Ouyang 2008). Another approach is to linearize the compliance function then use optimality criteria methods to minimise structure volume or weight, subject to a probability constraint on the maximum compliance value (Logo 2007).

Most RBTO methods consider uncertainty in loading magnitude. Although, direction uncertainty has been modelled using independent orthogonal loads with zero mean (Mogami et al. 2006), or by a non-probabilistic convex set model (Kang & Y. Luo 2009). Material property uncertainty has been limited to Young's modulus as a single uncertain variable affecting the entire structure equally (Kharmanda et al. 2004; Jung & Cho 2004; C. Kim et al. 2006). Uncertainty in non-structural mass has also been considered for truss structures when there is a reliability constraint on the fundamental eigenvalue (Mogami et al. 2006). When using an element based method, the thickness of elements has also been considered as a single uncertain parameter (C. Kim et al. 2006). Most of these uncertainties are either related to the load vector or are simple scalars on the stiffness matrix. This allows for reasonably straightforward computation of reliability constraint sensitivities. To the author's knowledge, more complex uncertainties that affect the stiffness matrix have not yet been considered in RBTO. These could include finer geometric uncertainties and a variation of material properties throughout the structure.

2.4.2 Robust topology optimisation

The main alternative to RBTO for including uncertainty in structural topology optimisation is to use a probabilistic objective, which is often referred to as robust optimisation. Most methods that adopt this approach focus on a probabilistic compliance objective. In this context there are two main problems studied: minimisation of the worst case scenario or minimisation of the average or expected compliance value.

The worse case scenario is a min/max problem, where the objective is to minimise the maximum possible compliance under the uncertain parameters. Loading conditions are usually considered as the uncertain parameters, which are modelled as a finite set of separate scenarios. The min/max problem with numerous loading scenarios can be efficiently solved using the semi-definite programming approach (Ben-Tal & Nemirovski 1997; Gournay et al. 2008). The number of scenarios that must be considered can be large if a large number of small incidental loads are used to ensure a robust solution is obtained. This approach can be inefficient, especially as some of the load scenarios can be non-critical to the robustness of the structure.

The cascading approach was developed to identify the most critical load scenarios that must be considered to produce a robust design, avoiding the unnecessary non-critical cases (Kocvara et al. 2000). The method was first to solve the problem using a small set of original loading scenarios. An eigenvalue analysis was then performed on the inverse stiffness matrix of the solution to identify the most critical loading scenarios. These are then added to the original set of loading scenarios and the optimisation continues with an increased number of scenarios. This process continues until the most critical new loading scenario for a solution does not significantly increase compliance. The worse case approach can produce robust structures, but has only been considered for loading uncertainty modelled by finite number of scenarios.

The minimisation of expected compliance problem can be naturally transformed into a multiple load case problem if loading uncertainties are defined as separate scenarios. The transformed problem is often called a stochastic programming problem, which can be easily solved for truss (Christiansen et al. 2001; Evgrafov & Patriksson 2003; Alvarez & Carrasco 2005) and continuum structures (Conti et al. 2008). However, if loading uncertainties are defined as continuous probabilistic variables, then discretization is required to use the stochastic multiple load case approach. Discretization can be achieved by simple sampling, where an upper bound on the accuracy of the approximation can be computed (Calafiore & Dabbene 2008). Alternatively, a quadrature rule can be used to integrate continuous probability density functions (Evgrafov & Patriksson 2003; S. H Lee et al. 2009). The quadrature points then define the scenarios or load cases used when approximating expected compliance. However, it has also been suggested that some situations allow for an explicit analytical computation of expected compliance (Calafiore & Dabbene 2008). This seems attractive, as it guarantees accuracy and may be more efficient to solve when there are a large number of uncertain variables. However, this has received little interest in the literature.

Uncertainty of nodal locations in a truss ground structure has also been considered when solving the minimisation of expected compliance problem (Guest & Igusa 2008). This novel approach models nodal location uncertainty using small equivalent uncertain loads. This avoids adding uncertainty to the stiffness matrix and instead deals with an equivalent and simpler multiple load case problem. Robust topology optimisation has also been applied to design compliant mechanisms, where the objective was to maximise the mean output displacement whilst minimising its variance under input loading uncertainty (Kogiso et al. 2008). The statistical moments were computed using a first order approximation, where the mean value is simply computed using mean loading conditions and variance is approximated using the derivative of displacement with

respect to the uncertain loads. Compliance variance can also be approximated using the quadrature scheme (S. Chen et al. 2010).

In robust topology optimisation, loading uncertainty has mainly been considered using a finite set of scenarios, either directly defined or derived from a discretization technique. Therefore, uncertainty in loading direction is either introduced using additional orthogonal load cases (Alvarez & Carrasco 2005; Kogiso et al. 2008) or by discretisation of a continuous distribution (Evgrafov & Patriksson 2003; S. Chen et al. 2010). Beyond loading uncertainty, the quadrature approach can be used to discretize more general uncertainties by defining a number of different scenarios with different loading and possibly different material properties.

Robustness is often defined as insensitivity to uncertainties (Doltsinis & Kang 2004; G-J. Park et al. 2006). Thus, it is important to consider the variance of an objective function to obtain a robust design. However, most of the methods discussed above only consider the expected compliance, which can produce a design with good average performance, but does not necessarily guarantee insensitivity to uncertainties or a low compliance variance. The importance of including variance has been discussed by some authors (Alvarez & Carrasco 2005; Guest & Igusa 2008), but it is a harder more complicated problem to solve compared to minimisation of expected compliance. Variance has only been considered in an approximated sense using quadrature discretisation (Chen et al. 2010) or a first order approximation (Kogiso et al. 2008).

2.4.3 Conclusions

Uncertainty has been introduced into structural topology optimisation using two main approaches: RBTO and robust optimisation. The RBTO approach includes uncertainties as constraints on failure probability. This approach has been

implemented efficiently and stably using FORM, PMA and the single loop technique. The objective is often to minimise weight and failure states have been defined using critical displacements, fundamental eigenfrequency and stiffness. The uncertain parameters considered include loading, Young's modulus, thickness of a two dimensional structure and non-structural mass.

The robust optimisation approach introduces uncertainty into the objective function, which can become either a worse case scenario or average performance. In the context of topology optimisation this approach has mainly focussed on a compliance objective with uncertainty in loading conditions. Loading uncertainty is usually defined as either a finite set of loading scenarios, or a discretization of continuous uncertain variables. For the continuous case, in some situations an analytical approach could be adopted that would avoid approximation and may improve efficiency compared to discretisation. Including variance into the robust objective function is important to produce a solution invariant to uncertainties. However, the analytical approach and the inclusion of variance have received little attention to date.

2.5 Optimisation methods

This section briefly reviews the main aspects and methods used to solve optimisation problems. An optimisation problem can be defined as minimising an objective function whilst satisfying all the equality and inequality constraints (Haftka & Gürdal 1992; Arora 2004; Rao 2009). Thus, the aim of optimisation methods is to find the design, characterised by a set of design variables, that solves this problems. Numerous optimisation methods have been developed to solve a wide variety of problems. This brief review is limited to the fundamental methods and aspects of optimisation.

2.5.1 Linear Programming

The linear programming (LP) method can be used to solve problems where the objective and constraints are linear or can be sufficiently approximated as being linear. Linear programming problems are usually stated in a standard form, where the equality constraints and design variables are all non-negative. When a LP problem is stated in this form it can be solved using the classic Simplex method (Dantzig & Thapa 1997). The principle of the Simplex method is that the optimal solution lies on the boundary of the design space at a vertex created by an intersection of the linear constraints. Starting with a feasible solution, the method simply searches through the vertices by looking for a neighbouring vertex with a lower objective value, until an optimal point is reached.

The Simplex method is simple and effective at finding optimal solutions to LP problems. However, it can required a large amount of computational resources, especially for problems with a large number of constraints and variables. One possible remedy is to decompose the original problem into smaller sub-problems that are solved almost independently (Dantzig & Wolfe 1960), although this is not always possible. The key limitation of the Simplex method is that it can only search along the boundary of the design domain as it moves between neighbouring vertices. This realisation led to the development of the interior method that allows search directions inside the domain (Karmarkar 1984). This approach can significantly reduce convergence time compared with the original Simplex method.

2.5.2 Non-Linear Programming

If the objective function and constraints cannot be expressed in explicit forms of the design variables, or are simply too complicated to manipulate, then numerical methods are required to solve the optimisation problem. Non-linear programming

(NLP) approaches are an important group of numerical optimisation methods. Most NLP approaches are iterative and follow this general algorithm (Rao 2009):

1. Start with an initial design.
2. Find a suitable search direction that points in general direction of the optimum.
3. Find an appropriate step length to move in that direction.
4. Update the design by moving the step length in the direction.
5. Test for optimality. End or goto step 2.

There are many NLP methods that can handle constrained and unconstrained problems and this review is limited to the more popular and fundamental approaches.

Unconstrained NLP methods

NLP optimisation problems with no constraints can be solved by direct search methods. Direct search methods include random search, where the search direction is randomly generated each iteration, grid search, where all objective values are computed at grid points of a suitable grid and exhaustive search where the entire design space is investigated to find the optimal design (Rao 2009). The advantages of these approaches are that they are simple, do not require gradients of the objective function and work even if the objective is discontinuous or non-differentiable. However, they often require a large number of objective function evaluations, which makes them inefficient.

There are several methods that are applicable for solving NLP problems with only one design variable. These univariate methods include elimination approaches that progressively reduce the design space until a small region containing the optimal point remains (Rao 2009). Newton's method can be used to solve univariate NLP problems if first and second order derivatives of the objective function can be computed. This is achieved iteratively by approximating the function using a second order Taylor series and finding the point where the first derivative is zero

to create the design for the next iteration. If derivatives of the function cannot be computed explicitly then the Quasi-Newton method can be used, where derivatives are approximated (Broyden 1967). Univariate NLP optimisation methods can be extended to solve multivariate problems by changing one variable at a time. This approach is simple and easy to implement, but is often slow to converge and tends to oscillate during optimisation (Rao 2009). This is mainly because the search direction is always parallel to the coordinate axes defined by the design variables. The search direction can be improved by performing simultaneous univariate steps, known as a pattern search direction (Powell 1964).

NLP methods that utilise derivatives or gradients of the objective function are known as descent methods, as the gradient provides information about the direction of steepest descent. The methods usually converge quicker compared with those that do not use gradients (Rao 2009). The method of steepest descent uses the negative of the gradient vector to update the design. This may seem efficient, but for most NLP problems the gradient is a local property and oscillation can occur during optimisation leading to slow convergence. The convergence rate of the steepest descent method can be improved by employing conjugate gradient method. Conjugate gradients improve the descent direction by using gradient information from the previous iteration (Fletcher & Reeves 1964). The extension of Newton's method to solve multivariate NLP problems can be seen as a descent method as it requires first and second order partial derivatives of the objective function. The second order partial derivatives form a matrix known as the Hessian, which has to be inverted each iteration to compute the search direction. However, this approach is not practical for problem with a large number of design variables or a complicated objective function due to effort required to compute and invert the Hessian matrix each iteration (Rao 2009).

Constrained NLP methods

Approaches for solving constrained NLP problems fall into two categories, direct and indirect methods. Direct methods include extending the direct search methods to include constraints, although this approach is again inefficient due to the large number of objective function evaluations required. The Sequential Linear Programming (SLP) method is more efficient, as it makes use of the Simplex method (Kelley 1960). In SLP the objective and constraints are approximated using a first order taylor series expansion about the current design variables and an improved design is obtained by solving the approximated problem by the Simplex LP method. This process is repeated until an optimal point is found.

Sequential Quadratic Programming (SQP) combines several important ideas in optimisation and is one of the most successful direct methods for solving constrained NLP problems (Boggs & Tolle 1995). The first idea is to transform the constrained problem into an equivalent unconstrained one by adding the constrains to the objective function using the Lagrange multiplier method. Next a set of simultaneous equations are derived using the Kuhn-Tucker optimality conditions for the equivalent Lagrangian objective function. The Kuhn-Tucker conditions state that the partial derivatives of the Lagrangian function with respect to each design variable and Lagrange multiplier has to be zero for a point to be a relative minimum. Finally the simultaneous equations are non-linear and are solved iteratively using Newton's method to arrive at an optimal design.

Indirect methods for solving constrained NLP problems generally employ the penalty function method to transform the problem into an equivalent unconstrained one (Zangwill 1967). The approach adds the constraints to the objective function using penalty factors that are reduced during the optimisation, such that the original problem is recovered when the solution converges. There are two main approaches for including penalty factors. The interior method adds penalty factors such that the constraints remain feasible throughout the

optimisation, whereas the exterior method starts with an infeasible design that becomes more feasible as the optimisation progresses. The penalty factor approach works with any unconstrained optimisation method. However, the efficiency of the approach can be dependent on the initial factor and how it is modified during optimisation.

2.5.3 Other methods

The NLP methods discussed above are efficient at finding optimum designs when the objective, constraints and design variables are reasonably continuous and the design space is convex, such that it only has one relative minimum point. If severe discontinuity exists the inefficient direct search methods have to be employed. Furthermore, if the design space is non-convex the optimisation may fail to find the global optimum, instead falling into a local minimum point. These two limitations arise because gradients are used to determine the search direction and hence are unable to handle discontinuities or escape a local minimum point.

Optimisation techniques that are conceptually quite different to NLP have been developed in more recent years to efficiently solve complex problems that involve discontinuity and non-convex design spaces. A feature of these methods is they tend to only require objective function values and not the gradients, which is an attractive quality for solving complex problems. There exist numerous alternative optimisation methods (Rao 2009). This review concentrates on just two concepts Genetic Algorithms (GA) and Simulated Anealing (SA) for illustration.

Genetic algorithms

GA techniques are conceptually based on Charles Darwin's theory of survival of the fittest and the algorithms make use of genetic features such as reproduction, crossover and mutation (Gen & Cheng 1997). They differ from traditional NLP

methods in a number of areas. First GA methods start with an initial set, or generation, of trial designs instead of a single design, which helps the optimisation avoid getting stuck at a local optimum. The design variables are represented by a string of binary variables. Therefore, GA is naturally applicable to solve problems with integer variables, but can also represent continuous ones by extending the number of binary variables. The objective function values are used to assess the "fitness" of the design, which is the central concept of GA.

Once the initial generation of designs have been analysed and objective values computed, a new generation of designs are produced using random parent design selection and crossover from the old generation and random mutations. Fitter designs with lower objective values are more likely to breed and pass on their "genes" or design features to the next generation. With each generation the population converges towards a solution and the algorithm is terminated when the standard deviation of the "fitness" of the population is with a specified tolerance, or the maximum number of generations has been reached. GA techniques are particularly applicable to solve problems with mixed continuous-discrete design variables and discontinuous non-convex design spaces and often find the global optimum design. If standard NLP methods are used for these type of problems, they are likely to be inefficient, computationally expensive and often find a local optimum near the starting point, that may not be the global optimum (Rao 2009).

Simulated annealing

Annealing is the process of slow controlled cooling of molten metal to obtain a internal structure with the lowest energy state. This concept can be introduced into optimisation using a temperature parameter that is controlled using the Boltzmann probability distribution (Laarhoven & Aarts 1987). At high temperatures there is a high probability of obtaining any energy state, but at low temperatures it is less likely to be at a higher energy state. This behaviour can be used in optimisation to control the convergence and avoid local optima. Simulated annealing achieves this

by potentially allowing an updated design with a higher objective function value to be accepted based on the current "temperature". When the temperature is high, in the early stages of optimisation, a higher objective value is more likely to be accepted. Whereas, near the optimum, the temperature is reduced and a higher objective value is less likely to be accepted. During optimisation new designs are generated in the vicinity of current design by randomly perturbing the design variables. The new design is accepted based on the change in objective value and the current temperature, which controls the probability of accepting an increased the objective value. This has the benefit of avoiding local minima, as the design can move in the opposite direction to the steepest descent. Also, as SA does not require derivatives it is applicable to problems with discontinuity. However, the rate of convergence and computational effort required is affected by the starting design, initial temperature and the cooling law used to reduce the temperature during optimisation.

2.5.4 Conclusions

There are numerous optimisation methods that have been developed to solve a wide range of problems. Optimisation problems can be characterised by several aspects such as the convexity of the design space and the behaviour of the objective function, constraints and design variables. Each of these aspects must be considered to select a suitable optimisation method that can find the optimum design in an efficient manner. For example, is the objective function and constraints are linear with respect to the design variables, then the optimisation problem can be efficiently solved using LP methods. If non-linearity exists, but the design space is convex and there is no severe discontinuity, then NLP methods are effective and efficient at optimising the design. However, if the design space is non-convex or severe discontinuity exists in the problem, then alternative methods such as GA or SA are required to find the global optimum design.

2.6 Conclusions

Topology optimisation provides the greatest potential for finding the best possible structure, as it is not dependent on the initial design. Structural topology optimisation methods started with trusses then progressed to more general continuum structures using element based methods, such as homogenisation and SIMP. However, these methods can suffer from numerical instabilities and inherently produce solutions with unclear "fuzzy" boundaries. These limitations have led to recent interest in boundary based methods.

Boundary representation with splines allows direct export to CAD programs, which is attractive. However, spline based methods are difficult to implement when there are large boundary movements and topology changes. A more popular approach is to use an implicit boundary representation that allows for natural topology changes. Structures defined using implicit functions can be optimised using the level set method where velocities are defined using shape sensitivity analysis. This direct approach has been successful in solving topology optimisation problems, but has some limitations. The approach can be inefficient and slow to find an optimum, it lacks the facility to create new holes, potentially leading to initial design dependent solutions, and it employs a fixed grid FEM for sensitivity computation, which can produce maximum errors along the boundary. Various alternative approaches have been proposed to alleviate these limitations. However, the direct approach is still attractive, due to the development of robust and efficient numerical procedures.

Level set based structural optimisation is an iterative process, requiring FEA each iteration to compute displacements, adjoint states and sensitivities. The analysis step is often the bottleneck in topology optimisation. Therefore, an efficient FEA method is desirable to produce a computationally efficient method.

However, there is a trade-off between accuracy and efficiency for the range of existing methods. The simplest and most efficient approach is the area-weighted fixed grid method. However, poor computation of sensitivities can destabilise the optimisation. It is therefore desirable to identify the cause of poor sensitivity computation to improve the accuracy of this efficient method. Another possible approach is to create a new fixed grid FEA method that has similar efficiency to the area-weighted method, but obtains more accurate sensitivities.

The solutions obtained using direct level set optimisation method can be dependent on the initial design because it lacks the facility to create new holes. Existing methods that address this issue either rely on topological derivatives, with an unclear link to shape derivatives, or are heuristic and sensitive to the choice of parameters. Thus, there is a need for a hole creation method that makes a meaningful link between boundary optimisation and hole insertion, without being sensitive to the choice of parameters.

Uncertainty is an important aspect to consider in design and optimisation and there are two main approaches applied to topology optimisation. RBTO uses probabilistic constraints for critical failure conditions. Robust optimisation includes uncertainty in a probabilistic objective function, such as average or worse case performance. The robust approach has focussed on the compliance objective with uncertainty in loading. However, loading uncertainty is often modelled as a set of discrete cases or by discretizing a continuous probability function. The discretization approach for continuous loading uncertainties can be inefficient if there are a large number of uncertainties and accuracy is important. Although, it has been suggested that an analytical approach may be adopted in certain situations. Furthermore, including variance is important to produce a design that is insensitive to uncertainties, but has only been considered in an approximate sense.

2.7 Research aim and objectives

The aim of this research is to introduce loading uncertainty into level set based structural topology optimization to efficiently produce robust solutions.

Considering the scope of this thesis and the conclusions of the literature review the following research objectives are identified.

1. To investigate and improve the sensitivities obtained by the area-weighted fixed grid method for boundary based optimisation methods.
2. To create an alternative, efficient fixed grid method that can improve boundary sensitivity computation compared to the area-weighted method.
3. To implement an efficient and stable level set based structural optimisation method that solves the compliance problem using the direct approach. The implementation will be validated using widely studied examples.
4. To create and test a hole insertion technique for the direct level method, that makes a meaningful link between boundary optimisation and hole insertion and is insensitive to the choice of parameters. The new technique is again validated against widely studied examples.
5. To use an analytical approach to efficiently solve the robust minimisation of expected compliance problem, when there is uncertainty in loading magnitude and applied direction.
6. To extend the robust compliance problem to include variance.

Chapter 3

An investigation of the area-weighted fixed grid finite element method

3.1 Introduction

Practical structural optimisation methods usually require some form of Finite Element Analysis (FEA) to compute the required response states and associated sensitivities. Level set based structural shape and topology optimisation methods require shape sensitivities computed along the boundary to derive the velocity function used to update the design (Allaire et al. 2004; M.Y. Wang et al. 2003). A traditional Lagrangian mesh fits elements exactly to the structure boundary and could be employed to compute shape sensitivities (2.13). However, as the structure is altered throughout the optimisation re-meshing is unavoidable and can become computationally expensive (S. Y. Wang & M. Y. Wang 2006a; Rong & Liang 2008).

To avoid expensive re-meshing, practical level set based optimisation methods usually employ a fixed Eulerian mesh to perform FEA and compute sensitivities.

The geometry of the fixed mesh, or grid, remains unaltered throughout the optimisation and design changes are reflected in the FEA by adjusting properties of individual elements. The structure is enclosed within the fixed mesh, thus the mesh is composed of two distinct regions: the real solid material part and a fictitious void part. Thus, three types of elements can exist within the fixed mesh: solid elements completely inside the structure (I elements), void elements completely outside the structure (O elements) and approximated elements intersected by the boundary (A elements), Figure 3-1. Throughout this work, the solid material is assumed to be homogeneous and linear elastic.

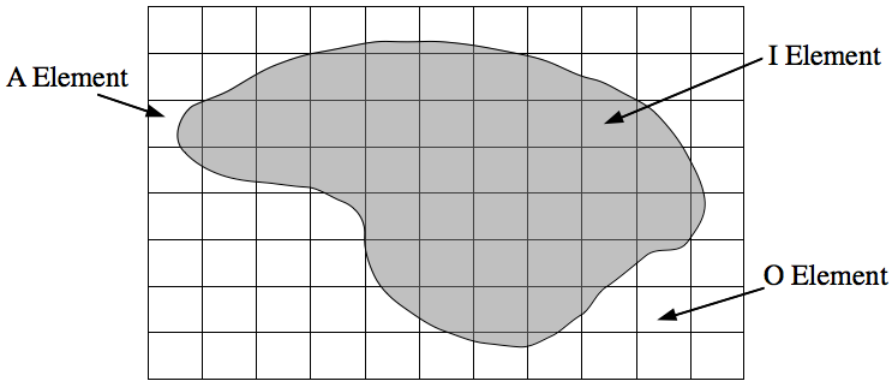

Figure 3-1: AFG element types.

3.2 Area-weighted fixed grid method

One approach to approximate the properties of intersected A elements is to weight their stiffness by the area fraction of real material within the element. Thus, stiffness matrices for A elements can be efficiently computed by simply determining area fractions. Stiffness matrices for the remaining elements in the fixed mesh are not approximated and can be pre-computed. Thus, computing the global stiffness matrix after a design update is a simple and efficient process. This Area-weighted Fixed Grid (AFG) method has been utilised by many level set based structural optimisation methods to perform FEA and compute shape sensitivities (Allaire et al. 2004; S. Y Wang et al. 2007; Z. Luo et al. 2007; J. Luo et al. 2008; Takezawa et al. 2010).

3.2.1 Formulation

There are two main approaches to treat the void part of the fixed mesh. The first is to fill it with a fictitious weak material with stiffness significantly lower than the real solid part (Allaire et al. 2004). Thus, Young's modulus at any point, x within the fixed mesh is defined by:

$$E^*(x) = \begin{cases} E & x \in \Omega_S \\ \gamma E & x \notin \Omega_S \end{cases} \tag{3.1}$$

where E is the Young's modulus of the real material and γ a small non-zero number (typically $\gamma < 10^{-3}$). The area-fraction weighted approximation of an A element stiffness matrix can be written as:

$$K_A = [(1-\gamma)\alpha + \gamma] K_I \tag{3.2}$$

where K_A is the stiffness matrix of an A element, K_I the matrix of an I element with geometry equal to the A element and α is the area-fraction of solid material within the A element. If the structure is defined by an implicit function (2.1), then the computation of α is straightforward:

$$\alpha = \int_{\Omega_e} H(\phi(x)) d\,\Omega_e \;/\; \int_{\Omega_e} d\,\Omega_e \tag{3.3}$$

where Ω_e is the domain of the element.

The alternative method for treating the void part of the fixed mesh is to remove it from the analysis, effectively setting $\gamma = 0$ in (3.1) and (3.2) and deleting O elements from the FEA. This approach is adopted throughout this work, however, it is approximately equivalent to using a small non-zero γ value. Adopting the second approach, the A element stiffness approximation in (3.2) becomes simply:

$$K_A = \alpha K_I \tag{3.4}$$

The real solid material is homogeneous and linear elastic. Thus (3.4) is equivalent to setting Young's modulus to αE for the approximated A element.

3.2.2 Discussion

The AFG method is simple and efficient, but has some notable drawbacks in the context of level set based optimisation. The approximation of element stiffness by the area fraction weighting does not allow for the FEA to take account of the precise boundary geometry, unless a highly dense mesh is used, which will increase the computational effort (S. Y. Wang & M. Y. Wang 2006a; Jang & Y. Y. Kim 2005). Furthermore, the accuracy of boundary sensitivity computation using the AFG method can be poor and significant numerical instabilities may occur (S. Y. Wang & M. Y. Wang 2006a; Wei et al. 2010).

When using the AFG method to analyse a simple structure, it has been shown that the approximation of A elements can cause errors in the finite element displacement field when compared to theoretical values (Garcia-Ruiz & Steven 1999). Also, maximum displacement errors are found to occur at the structure boundary where A elements exist. The same analysis revealed that errors in stresses are also present and are greater where A elements occur and maximum at stress concentrations. Furthermore, stresses computed by nodal averaging are not guaranteed to converge in the limit of mesh refinement. This suggests that errors in shape sensitivity (2.13) and, therefore, velocity (2.14) are also greatest along the boundary, where A elements are present. This is a concern for level set based structural optimisation methods, as boundary velocities are used to update the design, driving it towards an optimum solution. A potential cause for stress errors is the discontinuous material properties at the boundary edge where A elements are present (N. H. Kim & Chang 2005).

3.3 Element sensitivity investigation

The effect of A element stiffness approximation on sensitivity computation is further investigated in this work by considering an abstract element that is composed of a homogeneous linear elastic material. The sensitivities considered in this work are for the compliance objective function. If there are no body forces then the shape sensitivity can be computed by the product of the stress and strain fields (2.13):

$$\varsigma(u) = \tilde{E}_{ijkl}\varepsilon(u)_{ij}\varepsilon(u)_{kl} = \sigma(u)_{ij}\varepsilon(u)_{ij} \tag{3.5}$$

where $\tilde{E}_{ijkl}$ is the elasticity tensor, $\varepsilon(u)_{ij}$ the strain tensor under the displacement field u and $\sigma(u)_{ij}$ the stress tensor. Stress, strain and sensitivity values are calculated under some simple loading conditions. These values are calculated using the AFG method by considering a square abstract element and approximating its stiffness by setting Young's modulus to αE. The approximated sensitivity values are compared to theoretical values calculated for a rectangular abstract element with a Young's modulus of E and an area equal to the amount of real material contained within the approximated element. The abstract element is defined in Figure 3-2a and the three simple loading conditions considered are shown in Figure 3-2b-d.

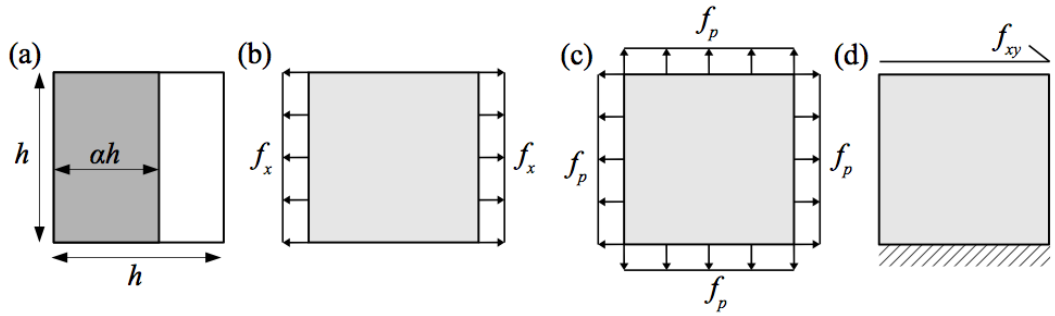

Figure 3-2: a) Abstract element, b) Uniaxial loading, c) Biaxial loading, d) Shear loading.

The relative error between approximated and theoretical sensitivities is computed by:

$$err(\varsigma) = |\varsigma - \varsigma_{AFG}| / |\varsigma| \tag{3.6}$$

where ς is computed by (3.5) using the equivalent rectangular element and real Young's modulus, E. Whereas ς_{AFG} is computed using a square element and approximated stiffness, αE.

3.3.1 Uniaxial loading

Stress, strain and sensitivity values are calculated for elements subject to uniaxial loading, Figure 3-2b. The relative error in sensitivity (3.6) is calculated by considering two rectangular elements with area equivalent to the real material within the approximated element; $\alpha h \times h$ and $h \times \alpha h$. The calculations are summarised in Table 3-1 and show that relative error in sensitivity value increases as α decreases. However, the magnitude of the error is dependent on the real geometry being approximated by the area-fraction weighting.

Young's Modulus	Dimensions	Stress (σ_{xx})	Strain (ε_{xx})	$\varsigma = \sigma_{xx}\, \varepsilon_{xx}$	err (ς)
αE	$h \times h$	f_x / h	$f_x / \alpha E h$	$f_x^2 / \alpha E h^2$	-
E	$\alpha h \times h$	f_x / h	f_x / Eh	f_x^2 / Eh^2	$(1 - \alpha)/\alpha$
	$h \times \alpha h$	$f_x / \alpha h$	$f_x / \alpha E h$	$f_x^2 / \alpha^2 E h^2$	$1 - \alpha$

Table 3-1: Uniaxial loading, sensitivity relative error calculation summary.

3.3.2 Biaxial loading

Stress, strain and sensitivity values are calculated for elements under biaxial loading, Figure 3-2c. In calculating the strain values the plane stress assumption is adopted, although similar results are obtained using plane strain. The calculations

are summarised in Table 3-2 and show that for Poisson's ratio values, $\nu < 1$, the relative error in sensitivity (3.6) increases as α decreases. However, the magnitude of the error is dependent on Poisson's ratio.

Young's Modulus	Dimensions	Stress (σ)	Strain (ε)	$\varsigma = \sigma\,\varepsilon$	*err* (ς)
αE	$h \times h$	$\sigma_{xx} = f_p/h$ $\sigma_{yy} = f_p/h$	$\varepsilon_{xx} = \varepsilon_{yy} = (1-\nu)(f_p/\alpha Eh)$	$2(1-\nu)\times (f_p^2/\alpha Eh^2)$	-
E	$\alpha h \times h$	$\sigma_{xx} = f_p/h$ $\sigma_{yy} = f_p/\alpha h$	$\varepsilon_{xx} = (\alpha-\nu)(f_p/\alpha Eh)$ $\varepsilon_{yy} = (1-\alpha\nu)(f_p/\alpha Eh)$	$(\alpha^2-2\alpha\nu+1)\times (f_p^2/\alpha^2 Eh^2)$	$(1+\alpha^2-2\alpha)/(1+\alpha^2-2\alpha\nu)$

Table 3-2: Biaxial loading, sensitivity relative error calculation summary.

3.3.3 Shear loading

The shear modulus, G is directly proportional to Young's Modulus, therefore, shear modulus using the area-faction approximation is αG. The stress, strain and sensitivity values are calculated for elements subject to a shear load, Figure 3-2d. The relative error in sensitivity (3.6) is calculated between an approximated element and two rectangular elements with area equivalent to the real material within the approximated element, Table 3-3. The two rectangles are again defined by: $\alpha h \times h$ and $h \times \alpha h$. For both cases the relative error in sensitivity value increases as α decreases, although the magnitude of the error is again dependent on the real geometry being approximated by the area-fraction weighting.

Shear Modulus	Dimensions	Stress (σ_{xy})	Strain (ε_{xy})	$\varsigma = \sigma_{xy}\,\varepsilon_{xy}$	*err* (ς)
αG	$h \times h$	f_{xy}/h	$f_{xy}/\alpha Gh$	$f_{xy}^2/\alpha Gh^2$	-
G	$\alpha h \times h$	$f_{xy}/\alpha h$	$f_{xy}/\alpha Gh$	$f_{xy}^2/\alpha^2 Gh^2$	$1-\alpha$
	$h \times \alpha h$	f_{xy}/h	f_{xy}/Gh	f_{xy}^2/Gh^2	$(1-\alpha)/\alpha$

Table 3-3: Shear loading, sensitivity relative error calculation summary.

3.3.4 Discussion

For all examples and loading conditions considered above, the relative error between approximate and theoretical sensitivities increases as α decreases. Although the magnitude of the error can depend on the geometry being approximated by the AFG approach, as for the uniaxial and shear loading cases, or Poisson's ratio, as for the biaxial loading case. However, the overall trend of errors increasing as α decreases suggests that boundary sensitivities, and thus velocities, may not be smooth between neighbouring A elements with significantly different α values, due to the varying levels of error in the approximation. Local roughness in the velocity function due to varying error in sensitivity computation can produce a spuriously rough boundary during the design update. This phenomenon may result in poor solutions or numerical instabilities.

3.4 Least squares fitting for the AFG method

Common methods for computing gradient fields in FEA are either to average element values at nodes and then interpolate using element shape functions, or to compute values at points within each element, then interpolate the sampled values using the least squares method (Cook et al. 2002). The second technique is based on the observation that stresses are often more accurate inside an element, especially at Gauss point locations. It has been shown that employing the weighted least squares technique to compute stresses when using the AFG method can produce superior results compared to the nodal averaging method (Garcia-Ruiz & Steven 1999). This section utilises the results from the element sensitivity investigation to develop a weighted least squares approach for computing boundary sensitivity values when using the AFG method.

3.4.1 The weighted least squares method

The basic principles of the weighted least squares method are presented in the context of interpolating sensitivities from a FEA. The aim of the least squares method is to minimise the difference between some known data points and the model used to interpolate between them. For example, a bilinear polynomial model could be used to interpolate between a set of n sampled sensitivity values, ς_i computed at points (x_i, y_i), where $n > 4$. The least squares method computes the unknown coefficients, c_j of the polynomial that best fit the sampled values by minimising the sum of the squared residuals between the model and sampled values:

$$\text{Minimise: } \sum_{i=1}^{n} \left(c_0 + c_1 x_i + c_2 y_i + c_3 x_i y_i - \varsigma_i\right)^2 \tag{3.7}$$

The weighted least squares method takes account of randomness or variability in the error between the sampled data values and the real system by introducing a weight for the residual in (3.7):

$$\text{Minimise: } \sum_{i=1}^{n} \left(c_0 + c_1 x_i + c_2 y_i + c_3 x_i y_i - \varsigma_i\right)^2 w_i \tag{3.8}$$

If the error between the sampled data and real system is large then it should be given less significance when solving the weighted least squares problem (3.8). Thus, if a sampled sensitivity value, ς_i is more likely to be erroneous then it is assigned a lower weight, w_i in (3.8).

3.4.2 Least squares parameters

There are various parameters involved in employing a weighted least squares method. Firstly, the sampled data are fitted to a model, which is usually a set of basis functions, such as a polynomial basis. The least squares method is then used to find the coefficients of the basis functions that best fit the sampled data (3.8). Therefore, an appropriate set of basis functions must be chosen so that the model is a good representation of the real system.

The type of finite element that forms the basis of the AFG method employed in this work is the plane four-node bilinear, or Q4 element (Cook et al. 2002). This element is also employed by other level set based structural optimisation methods (Allaire et al. 2004; S. Y. Wang & M. Y. Wang 2006a; S. Y Wang et al. 2007; Norato et al. 2007). For this element, there are two reasonable choices of where to sample sensitivities. The first option is to sample data at element centres and the alternative is to sample data at the Gauss points used to numerically integrate the element stiffness matrix (Cook et al. 2002). For a Q4 element, four Gauss integration points are required to accurately compute the stiffness matrix.

For some problems all data points are included in the least squares fit to compute basis function coefficients. However, this approach is not appropriate for computing sensitivities for a large problem with a complex geometry or sensitivity distribution. Therefore, the least squares method is only applied locally to compute sensitivity values at specific points of interest. This local modelling approach only uses a subset of the sampled data points within a specific support radius around the point of interest, when using the least squares method. The support radius must be large enough to include enough points to perform the least squares fit. For example if the basis functions describe a bilinear polynomial, at least four points are required to perform the fit (3.8). However, too large a radius may include too many points, leading to an increase in computational cost and possible loss of accuracy due to the influence of data far from the point of interest.

When using the AFG method it was demonstrated in the previous section that the error in sensitivity computation is likely to increase as the area-fraction of the element decreases. This suggests that it is beneficial to weight a sampled sensitivity by the area-fraction of the element used to compute the value. Furthermore, other research has shown that it is beneficial to weight sampled stresses by their inverse distance to the point of interest when using the least squares method (Garcia-Ruiz & Steven 1999). Thus, the compound weighting factor, w_i for each sampled sensitivity value used for the weighted least squares fit (3.8) is defined as:

$$w_i = \alpha_i / |\boldsymbol{x}_p - \boldsymbol{x}_i| \tag{3.9}$$

where $|\boldsymbol{x}_p - \boldsymbol{x}_i|$ is a measure of the distance between the location of the point of interest, $\boldsymbol{x}_p$ and the location of the sampled sensitivity value, $\boldsymbol{x}_i$ and α_i is the area-fraction of the element used to compute the sampled sensitivity.

The proposed weighted least squares method for computing boundary sensitivity values has three main parameters to investigate. Two polynomial models in two spatial dimensions (x, y) are considered, a bilinear (3.10a) and second order polynomial (3.10b):

$$\varsigma(x, y) = c_0 + c_1 x + c_2 y + c_3 xy \tag{3.10a}$$

$$\varsigma(x, y) = c_0 + c_1 x + c_2 y + c_3 xy + c_4 x^2 + c_5 y^2 \tag{3.10b}$$

where c_i are the unknown coefficients computed by the least squares fitting process. Two different sampling schemes are considered, one using sensitivity values computed at element centres, Figure 3-3 and the other using values computed at element integration points, Figure 3-4. Finally for each combination of polynomial model and sampling scheme a range of support radii are considered, where each radius is defined in terms of the fixed grid element edge length, h.

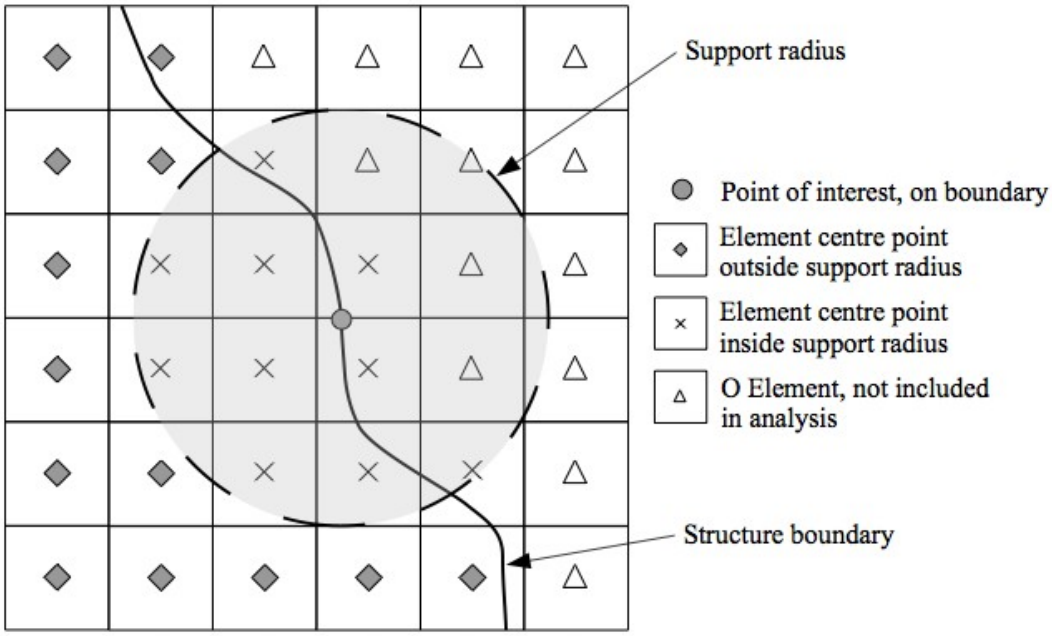

Figure 3-3: Element centre point sampling scheme for least squares fit with a support radius of 2*h*.

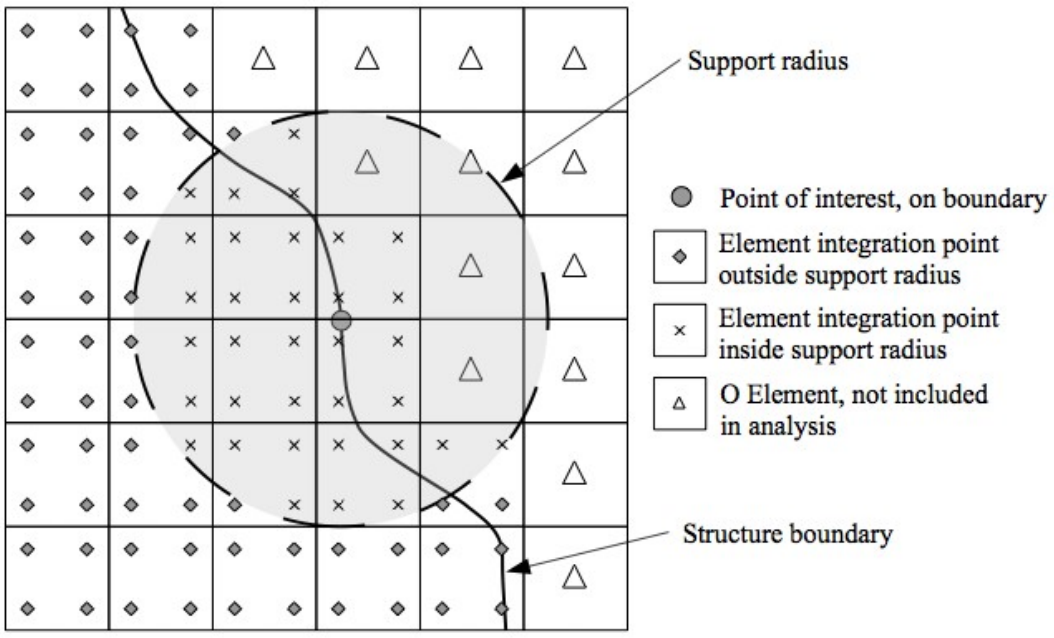

Figure 3-4: Element integration point sampling scheme for least squares fit with a support radius of 2*h*.

3.4.3 Least squares investigation

The aim of this work is to investigate the effect of the polynomial model, sampling method and support radius when employing the weighted least squares method to compute boundary sensitivity and velocity values using the AFG method. However, in the context of level set based optimisation, the relative distribution of sensitivities and not their exact values is important. This is because the velocity function (2.14) is usually normalised to meet the CFL stability condition (M. Y. Wang et al. 2003; Mei & X. Wang 2004). Therefore, in this investigation, boundary sensitivities are divided by the maximum absolute value to produce a distribution of normalised velocities.

The AFG method and weighted least squares technique are used to compute the distribution of normalised boundary velocities along the approximated boundaries of three example structures. Velocities are computed at fixed grid nodes that coincide with the boundary and at points where the boundary crosses the edge of a fixed grid element.

For comparison, velocity distributions are computed using the simple nodal averaging technique and a weighted nodal averaging method, where the relative contribution of each element sensitivity value is weighted by its area-fraction:

$$\varsigma_i = \sum_{j=1}^{n} (\alpha_j \varsigma_{i,j}) \;/\; \sum_{j=1}^{n} \alpha_j \tag{3.11}$$

where i is a point where averaging occurs, j denotes an element with an edge coincident with point i and $\varsigma_{i,j}$ is the sensitivity value for point i computed using nodal data from element j. For all examples, a fitted mesh is also constructed using Q4 elements of a similar size to those used for the fixed mesh. Normalised velocities are computed for the fitted mesh solution using the simple nodal averaging technique. The fitted mesh provides a reasonable reference solution to compare with the fixed mesh solutions.

Hole in plate example

The first example is a square plate with a central circular hole, subject to a tension load. Using symmetry conditions, only one quarter of the plate is considered. The distribution of normalised velocities is computed around the edge of the central hole. The fixed mesh is composed of 30 × 30 unit sized elements, Figure 3-5a, and the fitted mesh is shown in Figure 3-5b.

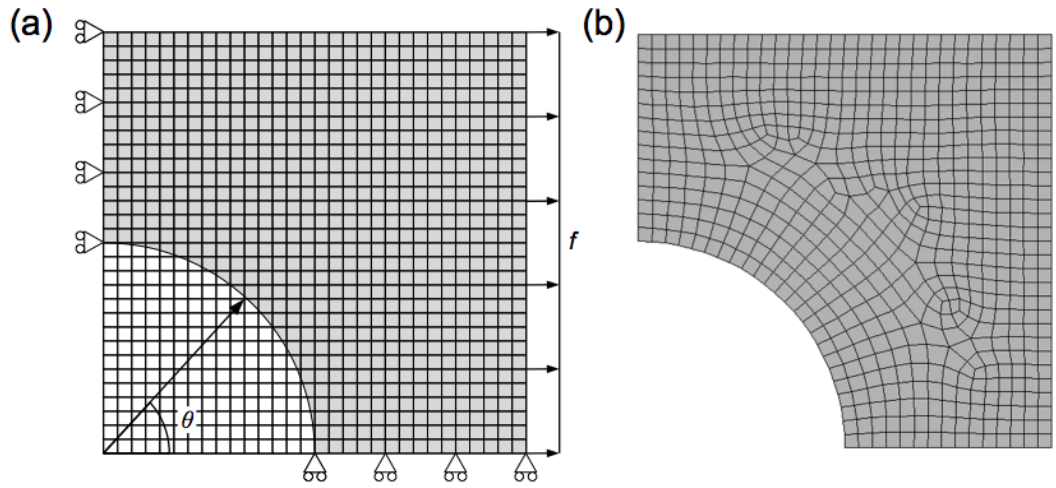

Figure 3-5: Hole in plate. a) Fixed mesh and boundary conditions, b) Fitted mesh.

The distribution of the normalised boundary velocity computed using the AFG method with the two nodal averaging techniques is shown in Figure 3-6. The area-fraction weighted averaging produces a smoother distribution with less sudden spurious fluctuations, when compared to the simple averaging method. This suggests that weighting sensitivity values by area-fraction is beneficial, even for the nodal averaging technique. However, the area-fraction weighted distribution is not as smooth as the solution obtained from the fitted mesh analysis. In general these observations also apply to the other examples considered in the remainder of this investigation. Therefore, the nodal averaging results are not shown for the remaining examples.

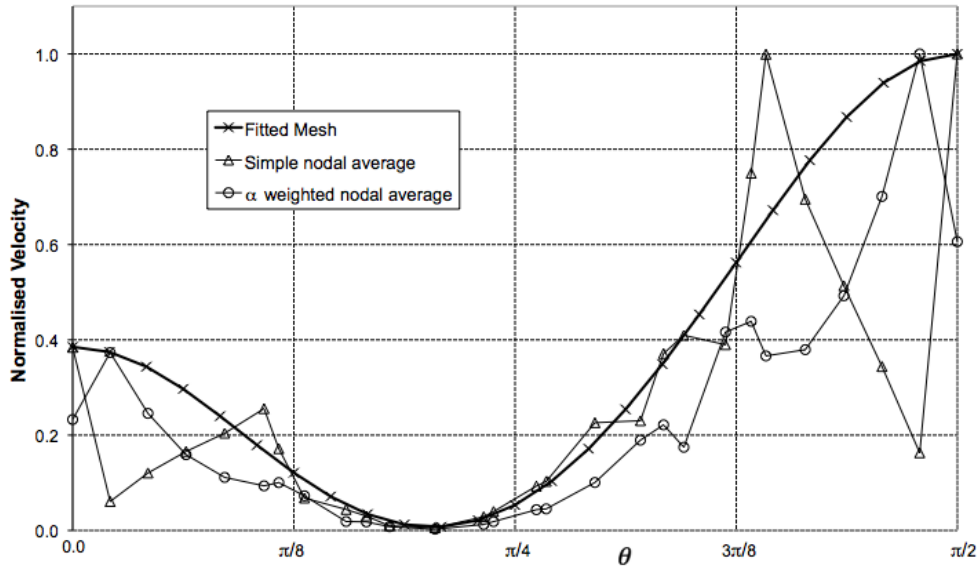

Figure 3-6: Hole in plate: Nodal averaged normalised velocity distributions.

The weighted least squares method is now employed to compute normalised velocities using the weighting function defined by (3.9). For brevity, only the results for the integration point sampling scheme with a second order polynomial are shown, Figure 3-7. However, the complete results reveal that as the support radius is increased a smoother distribution is obtained. For the largest radius considered, there was little difference between the two polynomial models and the two sampling schemes considered. Also, all distributions computed using least squares fitting were smoother than those computed using either of the direct averaging methods, Figure 3-6.

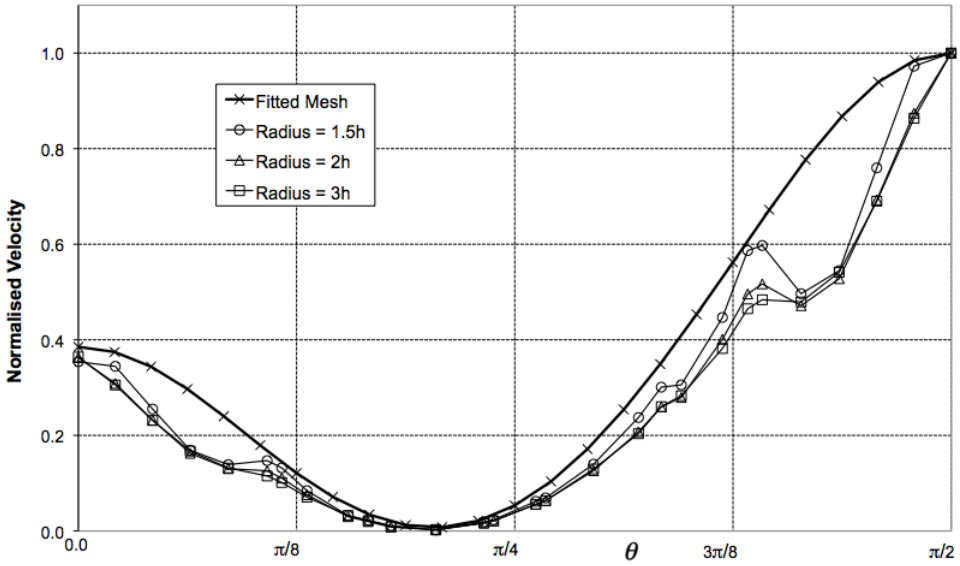

Figure 3-7: Hole in plate normalised velocity distribution. Weighted least squares method using integration point sampling and second order polynomial model.

Panel example

In this example a tapered panel is subject to a shear load (Cook et al. 2002) and normalised velocity distributions are computed along the top and bottom edges. The panel is analysed using a fixed mesh composed of square elements with edge length, h=4.0, Figure 3-8a, and a fitted mesh using elements of a equivalent size, Figure 3-8b.

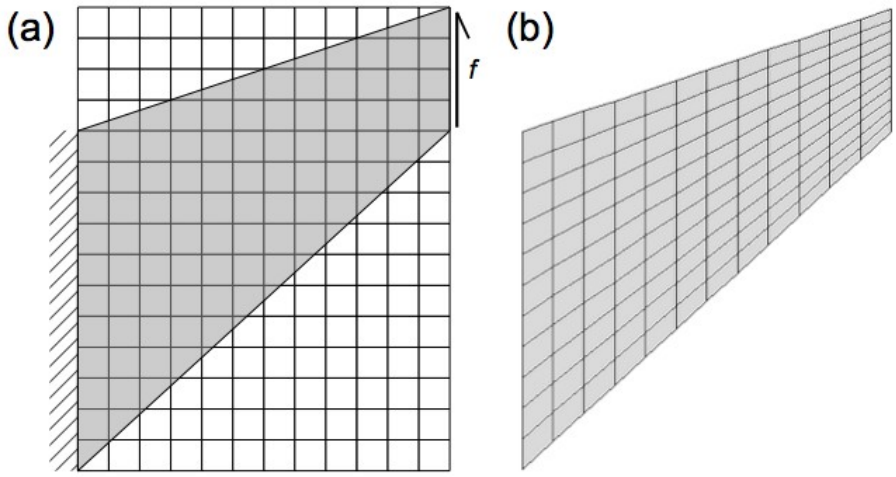

Figure 3-8: Panel example. a) Fixed mesh and boundary conditions, b) Fitted mesh.

Velocity distributions computed along the panel lower edge using the weighted least squares method reveal that, for this example, the support radius does not significantly influence the solution. Also, the choice of polynomial model does not greatly affect the solution, although distributions computed using the second order polynomial model are slightly smoother. Finally, the integration point sampling scheme generally produced smoother distributions compared with the centre point scheme, Figure 3-9, however, the difference is not significant.

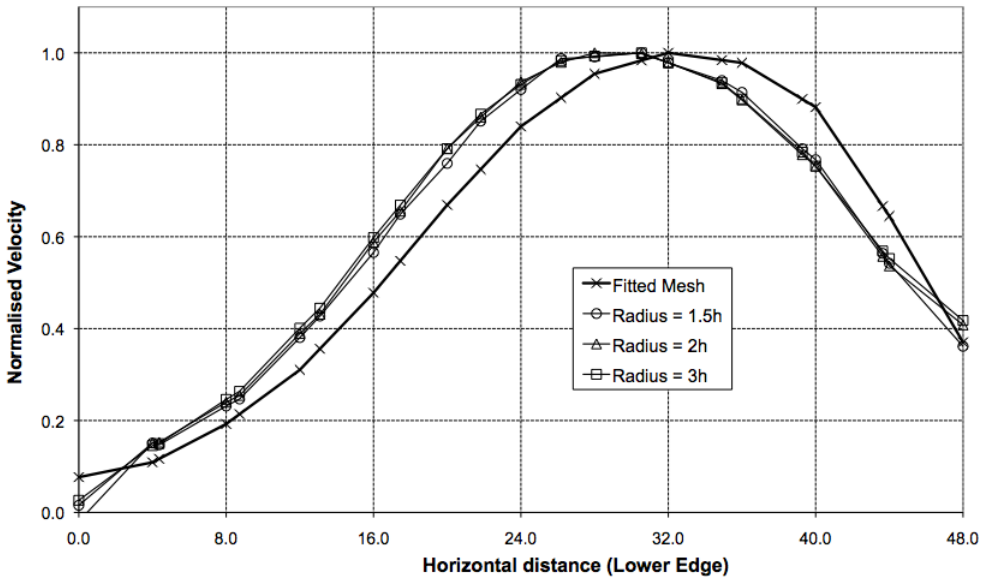

Figure 3-9: Panel lower edge normalised velocity distribution. Weighted least squares method using integration point sampling and a second order polynomial model.

Velocity distributions for the upper edge are now considered. In general, for this example, a larger support radius produces a smoother distribution, but this causes the distribution to drift away from the fitted mesh reference solution, Figure 3-10. However, this does not seem to occur for the integration point sampling method with a second order polynomial model, Figure 3-11. Furthermore, for both sampling methods, the second order polynomial produces smoother distributions closer to the fitted mesh reference solution.

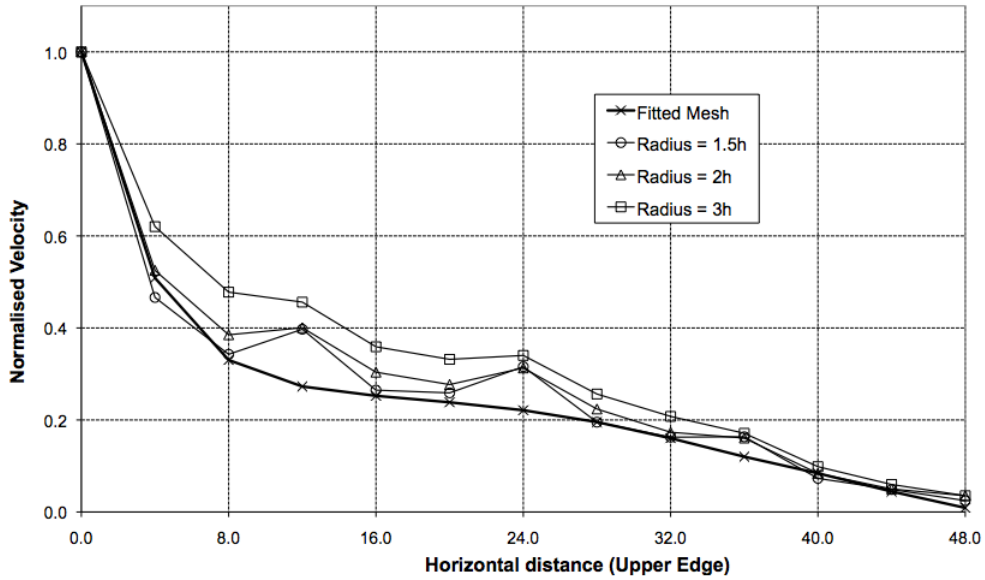

Figure 3-10: Panel upper edge normalised velocity distribution. Weighted least squares method using integration point sampling and a bilinear polynomial model.

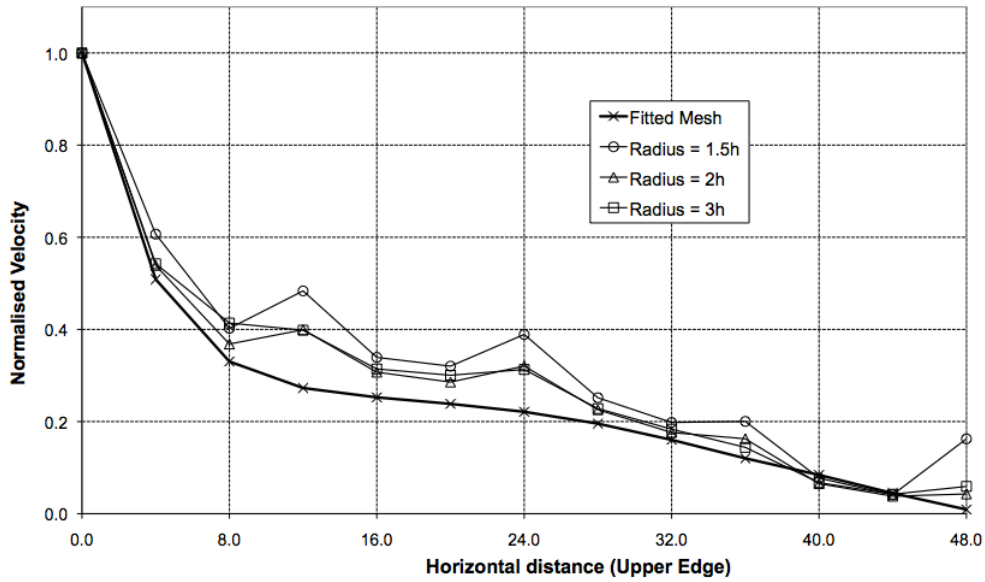

Figure 3-11: Panel upper edge normalised velocity distribution. Weighted least squares method using integration point sampling and a second order polynomial model.

Truss example

For this example a truss structure is analysed as a continuum using a fixed mesh, Figure 3-12a. The mesh is composed of 28 × 18 square elements with edge length h=0.5. Normalised velocities are computed along two edges where A elements occur in the fixed mesh, shown in Figure 3-12a. The fitted mesh used to compute the reference solution is shown in Figure 3-12b.

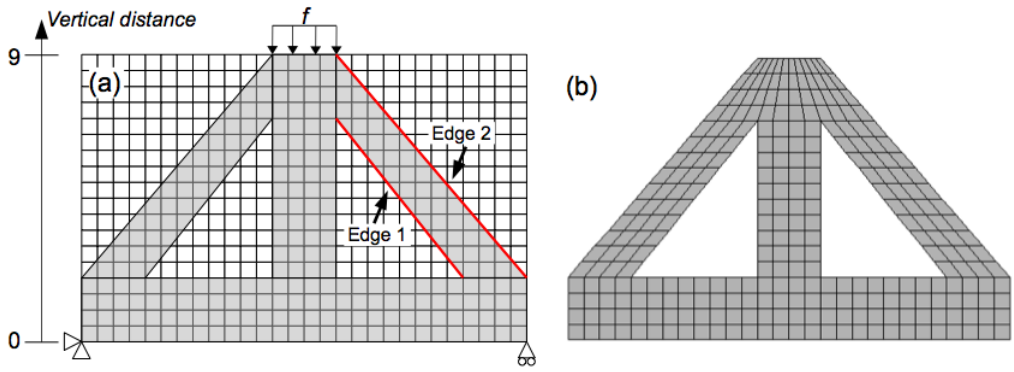

Figure 3-12: Truss structure. a) Fixed mesh and boundary conditions, b) Fitted mesh

The normalised velocity distributions computed along edge 1 (Figure 3-12a) of the truss are considered first. In contrast to previous examples, increasing the support radius does not improve the velocity distribution when using the weighted least squares technique. This behaviour can be attributed to sensitivities sampled in the central strut gaining more influence over the distribution computed along edge 1 as the radius is increased. Furthermore, this effect appears more severe for the bilinear polynomial model compared to the second order polynomial. Also, if only the second order polynomial model is considered, the results obtained using the integration point sampling scheme, Figure 3-14, appear closer to the fitted mesh reference solution, compared to solution computed using the central point scheme, Figure 3-13.

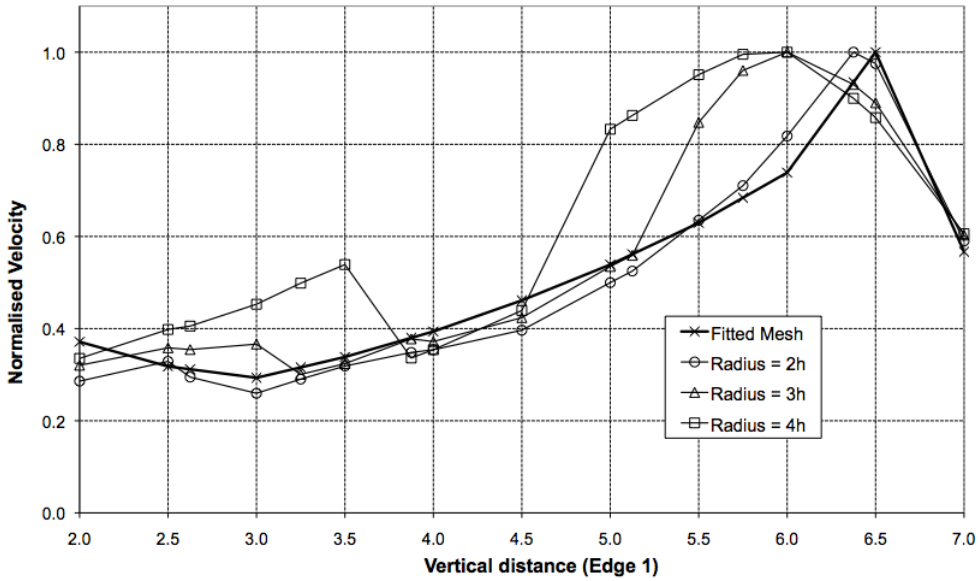

Figure 3-13: Truss Edge 1 normalised velocity distribution. Weighted least squares method using centre point sampling and a second order polynomial model.

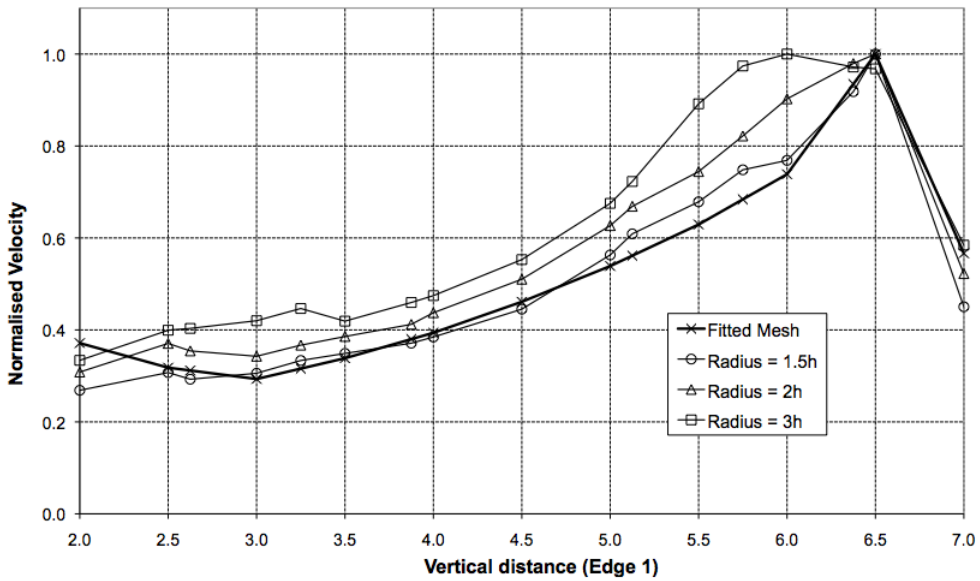

Figure 3-14: Truss Edge 1 normalised velocity distribution. Weighted least squares method using integration point sampling and a second order polynomial model.

For the normalised velocity computed along edge 2, the effect of the support radius is unclear, although smoother distributions can be obtained with a larger radius. The second order polynomial model generally produces superior results compared to the bilinear polynomial. However, the effect of the model was more significant for the centre point sampling scheme. In general the integration point sampling scheme produced more consistent results closer to the fitted mesh reference solution, Figure 3-15.

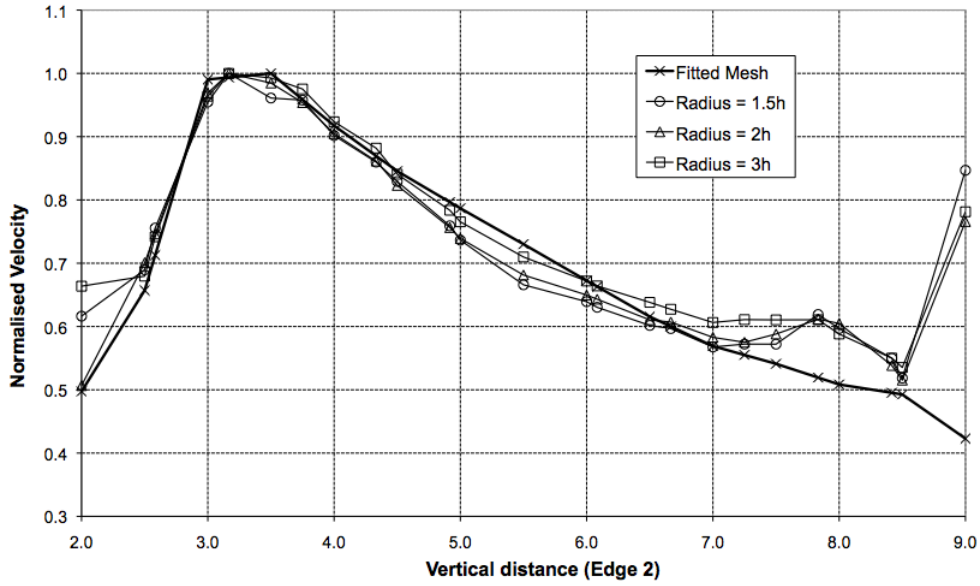

Figure 3-15: Truss Edge 2 normalised velocity distribution. Weighted least squares method using integration point sampling and a second order polynomial model.

3.4.4 Discussion

The results obtained for the AFG method using the nodal averaging techniques suggest that weighting the sensitivity values by area-fraction often produces a smoother distribution of normalised velocity along the boundary, compared to the simple averaging case. However, all distributions observed displayed fluctuations and often contained sharp changes that were not present in the fitted mesh reference solution. The results for the AFG method using the weighted least squares method demonstrate a significant improvement in smoothness of

boundary velocity distribution compared to the nodal averaged results. However, the choice of parameters used for the least squares method can significantly affect the solution.

The results of the investigation suggest the effect of support radius on the distribution of boundary sensitivities, or normalised velocities is situation dependent. For most examples, a larger radius results in a smoother velocity distribution. However, a larger radius can also include the influence of sampled sensitivities from parts of the structure not immediately connected to the point of interest. This may cause spurious variations in boundary velocity, as observed for edge 1 of the truss example (Figure 3-13, Figure 3-14). Therefore, it is suggested that a support radius in the middle of the range considered ($3h$ for centre point and $2h$ for integration point sampling) should be used, so that a reasonably smooth velocity distribution is obtained, without being significantly influenced by data from a different part of the structure.

For some of the examples considered in the investigation, the choice of polynomial model did not significantly influence the velocity distribution obtained. However, the second order polynomial model can produce superior results, especially when combined with integration point sampling. For example, the panel upper edge, Figure 3-11, and truss edge 1, Figure 3-14. This seems logical, as higher order models are more likely to produce superior results, as they can more accurately model distributions with increased complexity due to the inclusion of higher order terms.

For the examples considered, the integration point sampling scheme generally produces smoother results that are also closer to the fitted mesh reference solution. This could simply be a result of the increased number of sampled sensitivity values used to compute the interpolated values on the boundary. Although, more evidence is required to confirm this hypothesis.

To summarise, the weighted least squares method using an integration point sampling scheme with a second order polynomial model and support radius of $2h$ produced adequate or superior distributions of boundary velocity, compared with those obtained using other parameters. However, the investigation was restricted to the AFG method and a compliance objective function. Also, only one weighting factor was considered, equal to area-fraction divided by distance. Despite the limitations of the investigation, some of the observations should remain valid for other fixed grid methods and optimisation problems, such as the effect of support radius and the superiority of higher order models.

3.5 Conclusions

The AFG method is simple and efficient, and thus a popular approach for computing sensitivities in level set based structural optimisation. However, previous studies have found that errors in displacement and stress are greatest at the structure boundary. This is a concern for level set optimisation, as sensitivities computed at the boundary are used to define the velocity function that evolves the structure towards an optimum.

Considering a compliance objective, sensitivity values for an abstract element, subject to simple loading conditions, were computed using the AFG method and compared to theoretical values. The relative error between AFG and theoretical sensitivities was found to increase as the area-fraction decreased. Although, the exact trend and magnitude of the error was dependent on the loading conditions, the geometry approximated by the AFG method and potentially Poisson's ratio. However, the general trend between area-fraction and error suggests that the distribution of boundary sensitivities may not be smooth between neighbouring elements with significantly different area-fractions. This could produce spurious local fluctuations in the boundary velocity function, potentially leading to poor solutions and numerical instabilities.

A weighted least squares method was proposed to improve the distribution of boundary sensitivities when using the AFG method. Compliance sensitivity values were sampled at internal elements points and a series of local least squares fits were performed to compute the sensitivity and velocity values on the boundary. The weighting function applied to the sampled sensitivities was equal to area-faction divided by distance to the boundary point.

An investigation showed that smoother distributions of boundary sensitivity and hence velocity were obtained using the weighted least squares method, compared with two nodal averaging techniques. However, the parameters used for the least squares method can have a significant effect. A larger support radius often produced smoother sensitivity distributions. However, if the radius is too large, then data not directly connected to the boundary point can adversely influence the distribution. The choice between two polynomial models considered was often not significant, but, for some examples, superior results were obtained with a higher order model. For the Q4 element used in the investigation, sensitivities sampled at the four Gauss integration points often produced superior sensitivity distributions compared to the element centre point sampling scheme.

In conclusion, the weighted least squares method should be used in favour of nodal averaging when computing boundary sensitivities using the AFG method. Extensive investigations suggested that the integration point sampling scheme, a second order polynomial model and support radius of twice the element edge length is often a good choice of parameters.

Chapter 4

The boundary matching fixed grid method

4.1 Introduction

Boundary sensitivities computed using the AFG method can be improved by employing a weighted least squares technique. However, the AFG method is still unable to model the exact geometry of the boundary and errors in sensitivity computation can occur. Some fixed grid methods attempt to improve sensitivity computation by taking account of the exact boundary geometry when forming stiffness matrices (S. Y. Wang & M. Y. Wang 2006a; Jang et al. 2004).

In this chapter a new fixed grid method that models the boundary exactly is introduced and tested. The method is called the Boundary matching Fixed Grid (BFG) method and it differs from previous work as it fits shape functions to the exact boundary geometry without employing degenerate elements. The method is developed for two dimensional structures and applied to compute boundary shape sensitivities for the compliance objective function (1.1).

4.2 The boundary matching fixed grid method

The central aspect of the BFG method is to replace approximated finite elements cut by the boundary with elements that match the boundary exactly. The structure is approximated by the assembly of all elements completely within the boundary (I elements) and all elements cut by the boundary (A elements). The BFG method is reasonably efficient because only the stiffness matrices for elements cut by the boundary need to be computed each iteration. For two dimensional analysis, limiting one cut per element, three types of A elements are can be identified for the BFG method depending on how they are cut: triangular, quadrilateral and pentagonal, Figure 4-1.

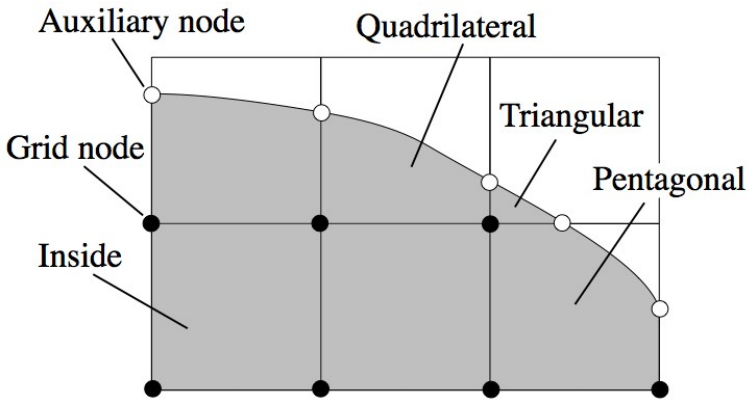

Figure 4-1: Boundary matching FG mesh

In order to assemble the global stiffness matrix, auxiliary nodes are required at points where the boundary intersects a grid line. There are therefore two sets of nodes, those used to construct the original fixed grid and those used for the analysis, Figure 4-1. The analysis set of nodes includes all fixed grid nodes that lie within the boundary and the required auxiliary nodes that lie on the boundary. The auxiliary nodes do not change the original fixed grid and are only required for analysis.

4.2.1 Triangular and quadrilateral elements

Element stiffness matrices for triangular and quadrilateral elements are formulated as the classic Constant Strain Triangle (CST) and isoparametric 4 node bilinear (Q4) elements, respectively (Cook et al. 2002). Triangular and quadrilateral stiffness matrices can be efficiently computed using algebraic expressions derived a priori avoiding the need for numerical integration. This process is natural for the CST element as its stiffness matrix can be calculated directly from its nodal coordinates (Cook et al. 2002). In the context of the BFG method this is further simplified as triangular elements always contain one right angle. Thus, the stiffness matrix only depends on the two perpendicular edge lengths, the element orientation and the material properties. Quadrilateral elements in the BFG method always contain two right angles. Therefore, the algebraic expressions for the stiffness matrix also depend on two edge lengths, the element orientation and the material properties.

4.2.2 Pentagonal elements

Pentagonal elements require more effort, as five node elements are not generally required by classic FEA. The key ingredients required to create a finite element are the shape functions that describe the behaviour of the element. A five node bilinear element (P4) is created by fitting bilinear shape functions to a pentagonal shape using the least squares method. Traditionally one shape function is required for each node of the element. Thus, the P4 element requires five shape functions, N_i with the following form:

$$N_i(x, y) = c_{0,i} + c_{1,i}\,x + c_{2,i}\,y + c_{3,i}\,x\,y \tag{4.1}$$

where $c_{n,i}$ are coefficients to be determined and the origin of the coordinate system (x, y) is located at the pentagonal element centroid.

The coefficients of the five shape functions are computed by solving a least squares problem so that the resulting functions best fit the Kronecker delta function property. This property states that each shape function should equal one at its associated node and zero at all other nodes:

$$N_i(x_j, y_j) = \delta(i, j) = \begin{cases} 1, & i = j, \quad j = 1..5 \\ 0, & i \neq j, \quad i, j = 1..5 \end{cases} \tag{4.2}$$

where i and j indicate node numbers. Therefore, the least squares problem that is solved for the coefficients of a shape function can be stated as:

$$\text{Minimise} : \sum_{j=1}^{5} \left(N_i(x_j, y_j) - \delta(i, j) \right)^2 \tag{4.3}$$

The P4 shape functions derived using the least squares approach satisfy the partition of unity condition necessary to replicate rigid body motion (Liu 2003):

$$\sum_{i=1}^{5} N_i(x, y) = 1, \quad \forall (x, y) \in \Omega_e \tag{4.4}$$

where Ω_e is the domain of the element. Furthermore, the linear field replication condition is also satisfied, which is necessary for the element to pass a standard patch test (Liu 2003):

$$\sum_{i=1}^{5} N_i(x, y) \varphi_i = \varphi(x, y), \quad \forall (x, y) \in \Omega_e \tag{4.5}$$

where $\varphi(x, y)$ is a field quantity interpolated between nodal values φ_i. However, because of the least squares fitting process the resulting P4 shape functions do not possess the Kronecker delta function property (4.2). This property allows simple application of boundary conditions (Liu 2003). This may not be a concern in the context of the BFG method employed for structural optimisation, as P4 elements usually exist on free boundaries where boundary conditions are not applied.

The P4 element stiffness matrix, K_{P4} is derived from the shape function polynomials in the classic manner:

$$K_{P4} = \int_{\Omega_e} B^T \tilde{E} B \, \Omega_e \tag{4.6}$$

where $\tilde{E}$ is the material property matrix and B is the strain displacement matrix derived from the shape functions and is defined as:

$$B = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} N_1 & 0 & \cdots & N_5 & 0 \\ 0 & N_1 & \cdots & 0 & N_5 \end{bmatrix} \tag{4.7}$$

Numerical integration is required to evaluate (4.6) over the irregular pentagonal element domain. The element is split into five triangular sub-domains and a second order mid-point rule is used over each sub-domain, Figure 4-2, which is sufficient for bilinear polynomials (Cook et al 2002).

$$K_{P4} = \sum_{i=1}^{5} (a_i/3)(K_{i,1} + K_{i,2} + K_{i,3}) \tag{4.8}$$

where a_i is the area of sub-domain i and K_{ij} is the value of the integrand in K_{P4} evaluated at edge mid-point j of sub-domain i. In practice only ten evaluations of K_{P4} are required to complete the numerical integration, Figure 4-2.

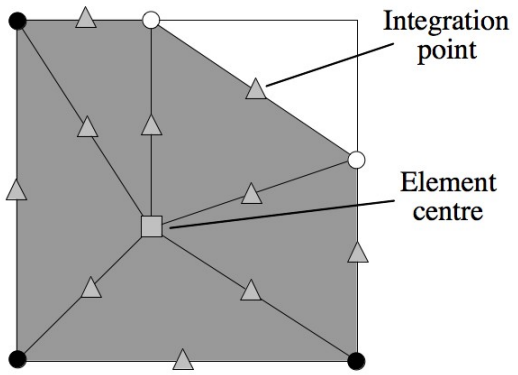

Figure 4-2: Pentagonal element integration points

4.2.3 Implementation

In general FEA distorted elements can artificially increase stiffness leading to reduced accuracy in computed displacement and stress values (Cook et al. 2002). Several distortions can affect the stiffness of the A elements employed for the BFG method including: nearly triangular quadrilaterals, high aspect ratio and highly skewed elements. Highly skewed elements are avoided by the BFG method as all elements contain right angles. However, high aspect ratio and near triangular quadrilaterals remain a concern.

To avoid similar meshing problems with the extended FEM, small local shifts of the boundary can be used to remove highly distorted elements (Daux et al. 2000; Van Miegroet & Duysinx 2007). This approach effectively deletes an auxiliary node if it is within the specified shift limit of a grid node inside the boundary, Figure 4-3. This "fit to vertex" method is utilised by the BFG method with a shift limit of 10% of the grid element edge length, h. The boundary shift is only necessary for analysis purposes and does not affect the real position of the structure boundary when employed for boundary based optimisation.

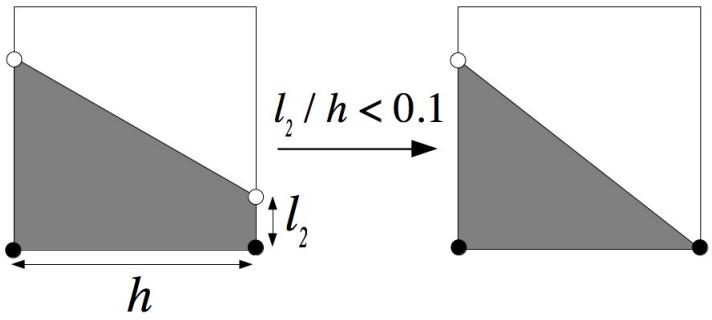

Figure 4-3: Fit to vertex method: an auxiliary node is deleted if within the shift limit.

In the two dimensional BFG method, A elements are restricted to intersection by a single part of the structure boundary. However, in boundary based optimisation, the situation where an A element is intersected by two parts of the boundary can arise. Thus, a method is required to rectify the situation so that the

element is only cut once. This can be achieved by simply assuming that one of the element grid nodes that does not lie on the boundary, does lie on the boundary. The node that is modified is chosen to minimise the change in boundary location compared to its actual position. Thus, the node that is closest to the boundary is assumed to lie on the boundary for the analysis, Figure 4-4. Again this small local shift is only applied for the analysis and does not alter the real position of the boundary.

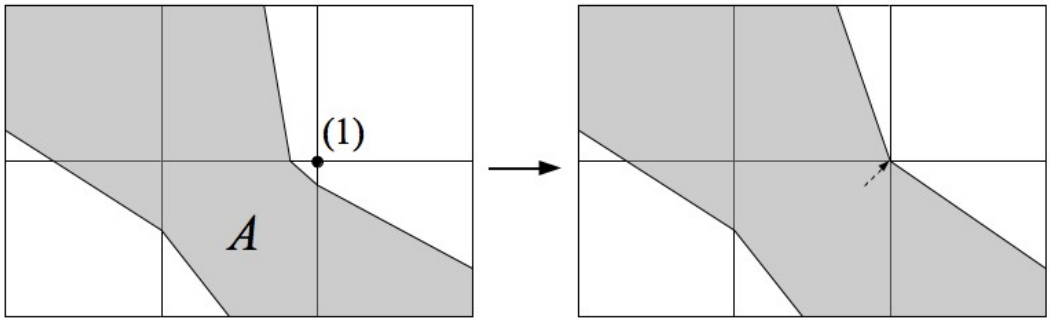

Figure 4-4: Element *A* is cut twice by the structure boundary. Node (1) is closest to the boundary and is assumed to lie on the boundary for the analysis (right picture).

4.3 Sensitivity computation investigation

The goal of the BFG method is to improve the accuracy of boundary sensitivity computation, compared to the AFG method. In the context of this work, boundary shape sensitivities (3.5) and normalised velocity distributions are computed for a compliance objective function (1.1). The three examples from Chapter 3.4.3 are again used for the investigation. However, only a representative set of results is presented from the complete investigation to highlight the key conclusions. The distributions shown are mostly for edge 2 of the truss structure, which is modelled as a continuum, Figure 3-12.

4.3.1 Nodal averaging method

A normalised velocity distribution is obtained from sensitivity values computed using the simple nodal averaging technique, Figure 4-5. The distribution is significantly different to the fitted mesh solution computed using the same simple nodal averaging method. The distribution contains two large peaks that correspond to the two smallest A elements along the boundary, as shown in Figure 4-6. This suggests that the large difference in neighbouring element sizes is causing an artificial disturbance in the local strain field (Cook et al. 2002), which adversely affects the accuracy of the computed sensitivities. A possible remedy is to use an area weighted nodal averaging technique to compute sensitivities:

$$\varsigma_i = \sum_{j=1}^{n} (a_j \varsigma_j) \;/\; \sum_{j=1}^{n} a_j \tag{4.9}$$

where i denotes a point between elements where averaging occurs and j denotes an element with area, a_j and an edge coincident with point i.

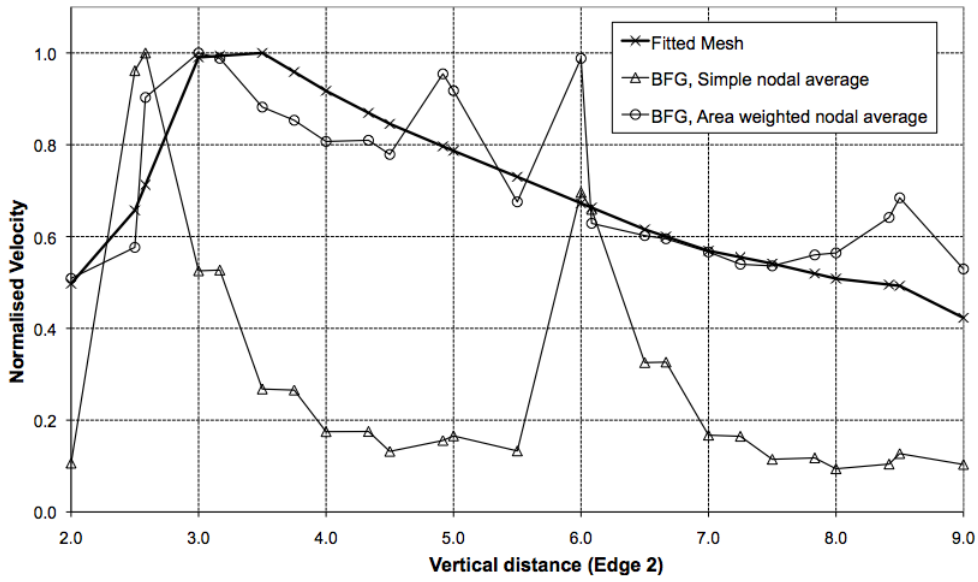

Figure 4-5: Truss edge 2, normalised velocity distributions computed using nodal averaging

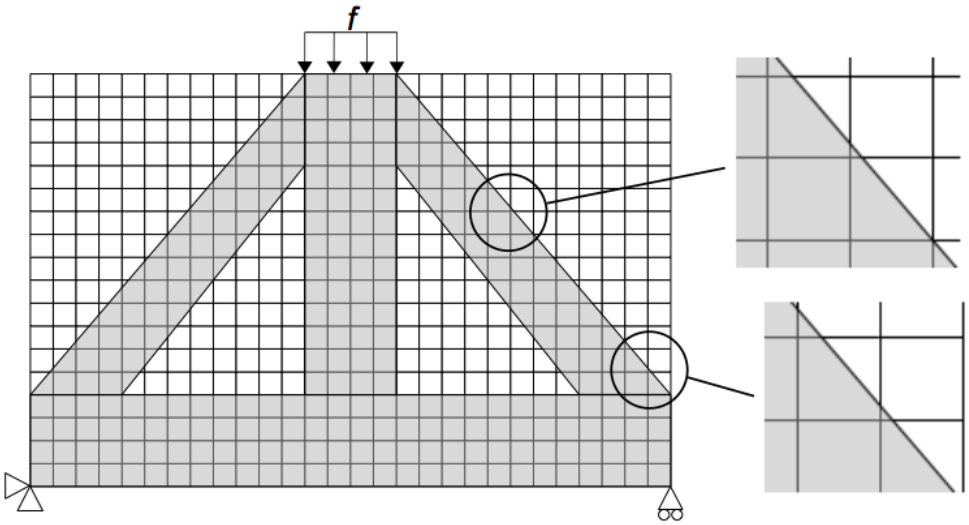

Figure 4-6: Truss structure, detail of small triangular elements along edge 2.

The normalised velocity distribution computed using the area weighted method shows an improvement over the simple averaging technique, Figure 4-5. However, peaks still occur where the small elements exist, although they are significantly reduced when using the area weighted averaging method (4.9). Furthermore, two additional peaks occur where P4 elements exist along the boundary. These peaks could be the result of spurious zero energy modes that occur due to the rank deficiency of the P4 element stiffness matrix (Cook et al. 2002). The rank deficiency arises from the least squares fitting of bilinear shape functions to a five node element. Thus, the P4 element currently employed by the BFG method has a rank of five, where it should have a rank of seven, allowing for three zero energy rigid body motions.

4.3.2 Weighted least squares method

The weighted least squares method investigated in Chapter 3.4 is found to improve the normalised velocity distribution for the AFG method, compared to nodal averaging techniques. A similar method is applied to the BFG method,

where the weighting factor is:

$$w_i = a_i / |\boldsymbol{x}_p - \boldsymbol{x}_i| \tag{4.10}$$

where $|\boldsymbol{x}_p - \boldsymbol{x}_i|$ is a measure of the distance between the location of the point of interest, $\boldsymbol{x}_p$ and the location of the sampled sensitivity value, $\boldsymbol{x}_i$ and a_i is the area of the element used to compute the sampled sensitivity. Element area is used in the weighting to counter the effect of neighbouring elements with significantly different sizes.

Normalised velocity distributions are computed using the centre and integration point sampling methods for edge 2 of the truss structure, Figure 3-12. Both sampling schemes are used with the second order polynomial model (3.10b) and mid-range support radius, as suggested in Chapter 3.4. For integration point sampling using the BFG method, triangular elements have one point located at the centre and only the five internal integration points are considered for a pentagonal element, Figure 4-2. The results show that, in general, the integration point sampling scheme produces a distribution closer to the fitted mesh reference solution, Figure 4-7. However, the peaks in sensitivity that coincide with pentagonal elements still persist for integration point sampling. Critically, where pentagonal shaped elements are present, neither distribution computed using the BFG method is as smooth and free of spurious fluctuations as the distribution computed using AFG, illustrated in Figure 4-7.

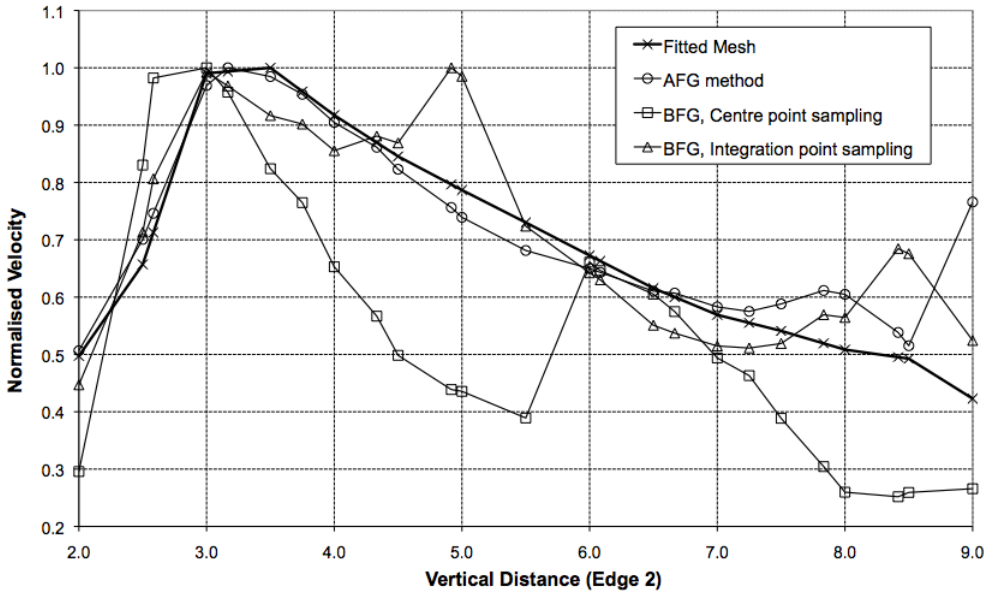

Figure 4-7: Truss edge 2 normalised velocity distributions using weighted least squares.

In general, the current BFG method does not compute superior distributions of normalised boundary velocity, when compared to the AFG method where P4 elements exist along the boundary. However, the BFG method can compute sensitivity values that are closer to the fitted mesh reference solution, Figure 4-8, although spurious peaks are still present for P4 elements. The upper edge of the panel structure example does not contain any pentagonal shaped elements, Figure 3-8. If sensitivity values are computed for this example, then spurious peaks do not occur and the values are much closer to the fitted mesh solution compared with those computed using the AFG method, Figure 4-9.

The BFG method shows potential improvement over the AFG method when the exact values of boundary sensitivities are important. This could occur if the internal part of the structure is updated simultaneously with the boundary, for example, by new hole insertion, (S. Lee & Kwak 2008). However, the issue of rank deficient P4 element matrices needs to be resolved to prevent spurious zero energy modes creating spurious local fluctuations in the boundary sensitivity distribution. This could be achieved by stabilising the element stiffness matrix

(Belystchko & Tsay 1983; Peters & Heymsfield 2003), or by using an alternative pentagonal element formulation, such as a variable topology element (Rashid & Gullett 2000) or a linear stress element (Peters & Heymsfield 2004).

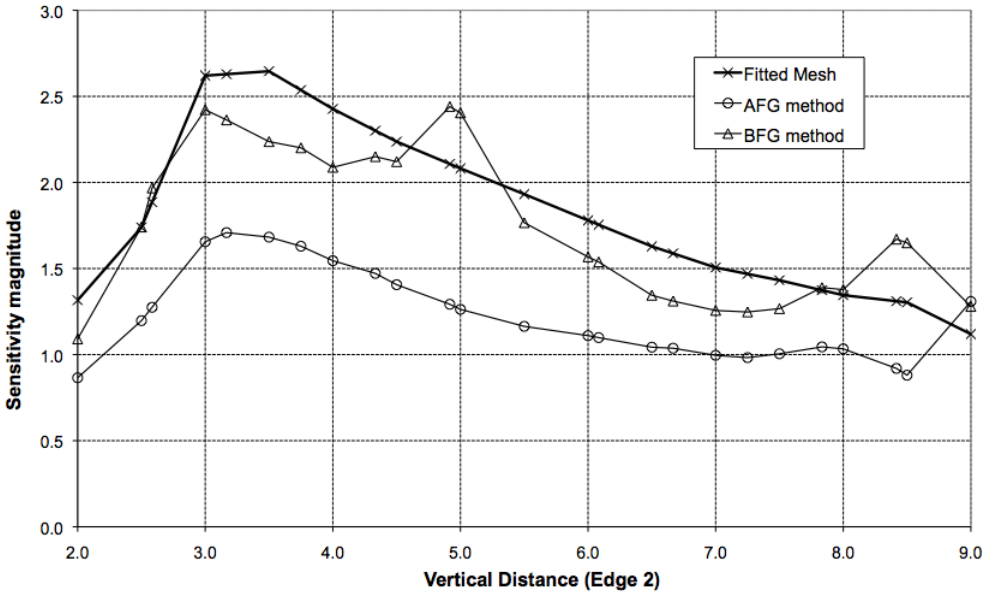

Figure 4-8: Truss edge 2 sensitivity magnitude. Fixed grid values computed using weighted least squares method with second order polynomial model, integration point sampling and a support radius of $2h$.

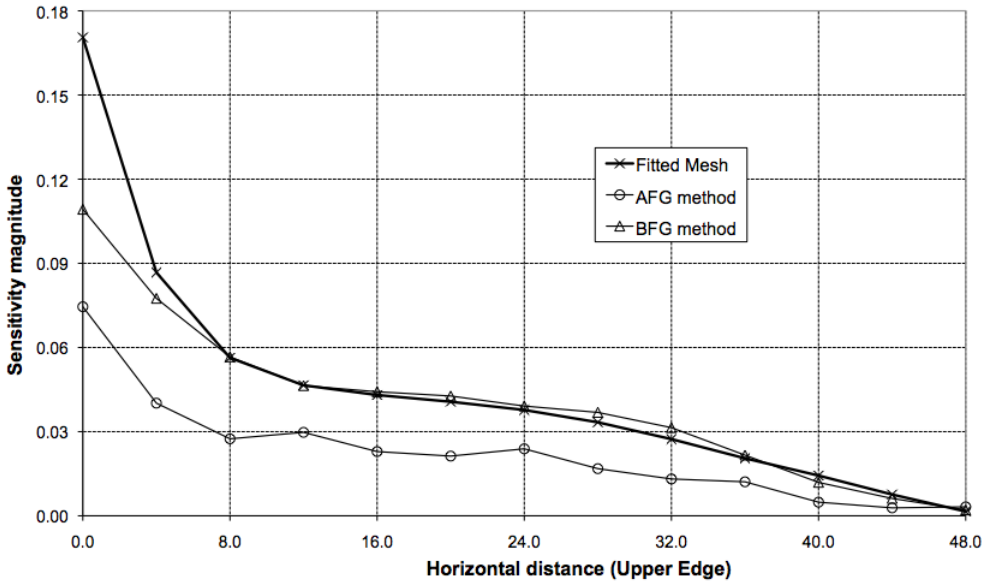

Figure 4-9: Panel upper edge sensitivity magnitude. Fixed grid values computed using weighted least squares method with second order polynomial model, integration point sampling and a support radius of $2h$.

4.4 Conclusions

The BFG method introduced in this work was employed to compute boundary sensitivity and normalised velocity values for a compliance objective. It is reasonably efficient because only stiffness matrices for elements cut by the boundary need to be computed each iteration. In FEA, the strain field between two adjacent elements of significantly different sizes can be artificially affected. This can cause artificial peaks and fluctuations in boundary sensitivity variation, thus, it was found beneficial to weight sensitivity values by the relative area of the element where they were computed.

Boundary sensitivity and normalised velocity distributions computed using the BFG method were further improved by employing a weighted least squares method. However, in general, the current BFG method did not compute superior distributions compared to the AFG method, when using the weighted least squares approach. This appears mainly due to artificial peaks in boundary sensitivity caused by spurious zero energy modes present in P4 elements.

Despite some short comings, the BFG method computes sensitivity values closer to those from a fitted mesh analysis, compared to the AFG method. This suggests that there is potential for improving sensitivity computation using the BFG method, although the spurious modes present in P4 elements remain a problem.

Chapter 5

Implementation of a level set based optimisation method

5.1 Introduction

The purpose of this chapter is to provide an overview of the direct level set based optimisation method used throughout this work. The various techniques and algorithms employed are discussed and critical details presented. The method is implemented to solve the minimisation of compliance problem for two dimensional linear elastic structures (1.1). Body forces are not considered throughout this work and are therefore omitted from the static equilibrium equation.

5.2 Algorithm overview

First a rectangular design domain, Ω is defined that encloses the initial structure and all possible solutions. The design domain is discretized by a grid of equal sized square elements. The initial structure, Ω_S is defined by an implicit function, $\phi(x)$ such that its zero level set coincides with the boundary:

$$\begin{cases} \phi(x) \geq 0, & x \in \Omega_S \\ \phi(x) = 0, & x \in \Gamma_S \\ \phi(x) < 0, & x \notin \Omega_S \end{cases} \tag{5.1}$$

The implicit function is discretized by defining values at domain grid nodes and interpolated using bilinear shape functions. The initial values of $\phi(x)$ are defined as a signed distance function, such that their sign is defined by (5.1) and their magnitude is equal to the distance from the grid node to the nearest boundary point. Discrete implicit function values are free to change throughout the optimisation. However, the structure has to remain inside the design domain, thus values along the edge of Ω are restricted to $\phi(x) \leq 0$ throughout the optimisation.

The structure is progressively optimised by updating the implicit function using a discrete Hamilton-Jacobi type equation as discussed in Chapter 2.3 (Allaire et al. 2004; M. Y. Wang et al. 2003):

$$\phi_i^{k+1} = \phi_i^k - \Delta t \, |\nabla \phi_i^k| \, V_{n,i} \tag{5.2}$$

where the velocity function is defined to be positive acting in from the boundary. The time step is defined by the CFL condition for stability:

$$\Delta t = \beta \, h / |V_n|_{max} \tag{5.3}$$

where $0 < \beta < 1$. A reasonably conservative value is chosen, $\beta = 0.5$, to ensure stability throughout the optimisation process. For the compliance minimisation

problem (1.1), the velocity function is derived from shape sensitivity analysis, as shown in Chapter 2.3 (Allaire et al. 2004):

$$V_n = \lambda - \tilde{E}_{ijkl}\,\varepsilon(u)_{ij}\,\varepsilon(u)_{kl} \tag{5.4}$$

The value of λ is the Lagrange multiplier for the volume constraint and is computed during each iteration. Details of the algorithm used for this computation are presented in Section 5.3.

The velocity function is initially computed just along the structural boundary. However, implicit function values are only defined at grid nodes. Thus the velocity function is extended to grid nodes by computing extension velocities, so that the update equation (5.2) can be applied.

The level set method can become unstable if the gradient of the implicit function becomes too flat or steep around the boundary (Allaire et al. 2004; S. Y. Wang & M. Y. Wang 2006a). Therefore, the implicit function is re-initialised to be a signed distance function at specific points during the optimisation.

The algorithm used to solve the minimisation of compliance problem (1.1) is illustrated in Figure 5-1. Further details for the various components of the algorithm are discussed below.

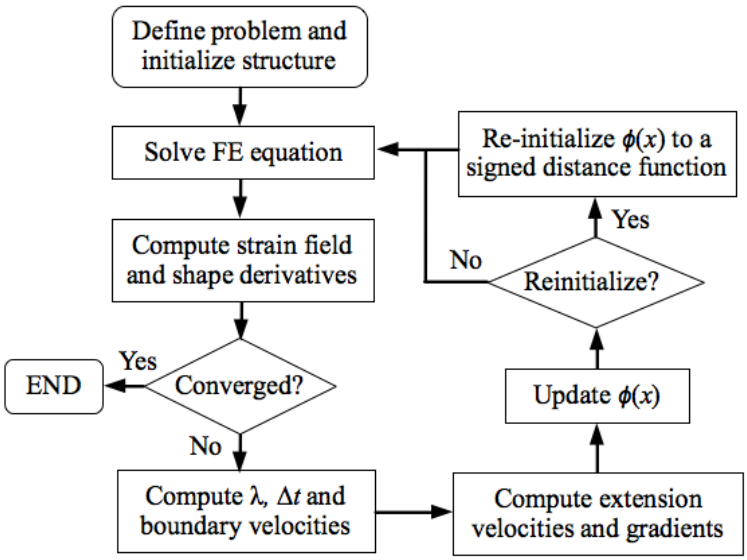

Figure 5-1: Level set optimisation algorithm

5.2.1 Sensitivity computation

The static equilibrium equation is solved efficiently each iteration using the AFG method discussed in Chapter 3. Boundary shape sensitivities used to derive the velocity function (5.4) are then computed using the weighted least squares scheme proposed in Chapter 3. For convenience, the grid used to discretize the design domain is also used for the AFG method. For simplicity, loads and essential boundary conditions are only applied at grid nodes. This alleviates the need to implement a method that can handle boundary conditions defined at locations other than grid nodes.

The definition of the structure by a discretized implicit function (5.1) allows for a straightforward implementation of the AFG method. Intersected elements in the fixed grid are easily identified by the sign of the implicit function at the element nodes. If an element has at least one positive and one negative nodal implicit function value then it is intersected by the boundary. The area-fraction of an intersected element is computed by determining the locations on the element edge where the implicit function is zero. The coordinates of these locations and nodes with a positive implicit function value form a polygon that defines the portion of the element lying within the structure. The area of the polygon is easily computed from its vertex coordinates and the area-fraction is simply the polygon area divided by the element area.

5.2.2 Extension velocities and re-initialisation

Extension velocities

Level set methods require the construction of extension velocities if the velocity function is not naturally defined at grid nodes (Sethian 1999). For the minimisation of compliance problem, velocities are derived from shape

sensitivities computed at the structural boundary. One possible method to construct extension velocities is to compute the strain field and shape sensitivities over the entire design domain by filling the void part with a fictitious weak material (Allaire et al. 2004; J. Luo et al. 2008) or by smoothing the field over the discontinuity at the boundary edge (S. Y. Wang & M. Y. Wang 2006). However, this simple method is unlikely to maintain the signed distance property of the implicit function and this adversely affects stability. Therefore, the implicit function must be reinitialised to a signed distance function frequently during the optimisation (Allaire et al. 2004).

To avoid frequent re-initialisation, an extension velocity technique designed to maintain the signed distance function is employed is this work (Adalsteinsson & Sethian 1999). The method ensures the preservation of the signed distance by using the fast marching method to solve the following equation for V_{ext}:

$$\nabla \phi_t \cdot \nabla V_{ext} = 0 \tag{5.5}$$

where ϕ_t is a temporary signed distance implicit function and V_{ext} is the extended velocity function. The extended velocity function is constrained to maintain the values already computed along the boundary. Boundary velocity values are computed where the boundary intersects grid elements and at grid nodes that coincide with the boundary. The temporary signed distance function, ϕ_t is computed from the current boundary position and is only used to computed the extension velocity in (5.5) at all grid nodes. The extension velocity is then used in (5.2) to update the primary implicit function, ϕ.

Narrow band approach

Further efficiency is gained by combining the extension velocity method (5.5) with the narrow band approach, so that extension velocities are only computed within a local region around the boundary and not over the entire domain (Chopp 1993; Adalsteinsson & Sethian 1995). This local region is fixed until the

boundary approaches its limits, then a new narrow band region is defined (Adalsteinsson & Sethian 1995; Sethian 1999). However, the signed distance function is only maintained within the narrow band. Therefore, before a new region is defined, the implicit function is re-initialised to a signed distance function over the entire domain.

The choice of width for the narrow band region can affect the efficiency of the method (Adalsteinsson & Sethian 1999). A smaller band width requires less computation of extension velocities, but more frequent re-initialisation. Thus, the overall computational cost is a function of extension velocity computations and number of re-initialisations required during the optimisation process. Numerical experiments showed that a band width of four grid lengths either side of the boundary was chosen as a reasonable computational trade-off between the two components, which is used for all examples in this Chapter.

Re-initialisation

Re-initialisation to a signed distance function can be performed by a number of methods. The straightforward, but inefficient approach is to explicitly compute the distance to the zero level set for each node (Chopp 1993). Another approach, used in several level set optimisation methods (Allaire et al. 2004; Mei & X. Wang 2004; S. Y. Wang & M. Y. Wang 2006a) is to iteratively solve the following equation:

$$\frac{\partial \phi}{\partial t} + sign(\phi^0)(|\nabla \phi| - 1) = 0 \tag{5.6}$$

where ϕ^0 is the current implicit function that is used as a starting point for the re-initialisation. However, this approach can lead to a movement of the zero level set and must be performed carefully to avoid difficulties (Adalsteinsson & Sethian 1999).

The re-initialisation approach adopted in this work is to use the fast marching method to solve the eikonal equation (Adalsteinsson & Sethian 1999):

$$|\nabla \phi_t| = 1 \quad , \quad \phi_t(x) = 0 \quad if \quad \phi^0(x) = 0 \tag{5.7}$$

The starting point of this approach is the zero level set, which is explicitly maintained during re-initialisation. The fast marching method is run separately for nodes inside and outside the structure. Initial signed distance values for nodes within one grid length, h of the zero level set are computed using distances to neighbouring intersections of grid lines and the zero level set. These values are then used as a starting point for the fast marching method. This process is similar to that used to compute ϕ_t for extension velocities (5.5).

5.2.3 Gradient calculation

The computation of implicit function gradient is the final component required to apply (5.2) and progress the structure towards an optimum. Gradient computation of a discretized function is usually performed numerically by a finite difference scheme. Level set methods often employ an upwind finite difference scheme that estimates the gradient based on the direction of movement of the implicit function. This promotes stability during the propagation of the boundary, as information travels in the direction of the boundary movement (Osher & Fedkiw 2003). The simplest upwind scheme in two dimensions simply selects either a first order forward or backward finite difference to estimate the gradient component in each direction based on the sign of the extension velocity:

$$|\nabla \phi_i| = max(sign(V_{n,i}), 0)\nabla_i^+ - min(sign(V_{n,i}), 0)\nabla_i^- \tag{5.8}$$

$$\nabla_i^+ = [\, max(D_i^{-x}, 0)^2 + min(D_i^{+x}, 0)^2 + max(D_i^{-y}, 0)^2 + min(D_i^{+y}, 0)^2 \,]^{1/2} \tag{5.9a}$$

$$\nabla_i^- = [\, min(D_i^{-x}, 0)^2 + max(D_i^{+x}, 0)^2 + min(D_i^{-y}, 0)^2 + max(D_i^{+y}, 0)^2 \,]^{1/2} \tag{5.9b}$$

where D_i^{+j} and D_i^{-j} are the finite difference operators for node i in direction j:

$$D_i^{+j} = (\phi_{i+1} - \phi_i)/h \tag{5.10a}$$

$$D_i^{-j} = (\phi_i - \phi_{i-1})/h \tag{5.10b}$$

where $\phi_{i\pm1}$ are implicit function values of adjacent nodes in the j direction.

Higher order finite difference schemes can be employed to improve the accuracy of the gradient estimation. The higher order Weighted Essentially Non-Oscillatory (WENO) scheme is third order accurate and can be up to fifth order in the smooth part of the solution (Jiang & Peng 2000). This scheme improves the robustness of gradient estimation and is employed to compute gradient components for (5.9) in this work.

The WENO scheme estimates a gradient component by constructing a set of three local polynomials from a stencil of six implicit function values around a node. This allows for a left or right bias when constructing the gradient estimate. The choice of bias is determined by the sign of the extension velocity according to the upwind scheme (5.8). The overall gradient estimate is the average of gradient values calculated from the polynomials, weighted by estimates of their relative smoothness. This approach can avoid discontinuities in the implicit function when estimating gradients and helps prevent noise being generated in the solution.

The polynomials used for the WENO scheme are constructed from first order finite differences contained within the stencil. However, near the edge of the design domain this stencil extends beyond the domain. Therefore, it is assumed that finite difference gradients for points beyond the design domain are equal to those at the nearest node on the domain edge. For example, the left biased stencil with backward finite differences is used to estimate the x-direction gradient for node i in Figure 5-2. However, two points lie outside the design domain, i-3 and i-2. Thus, the backward difference estimate for i-1 and i-2 is the same as node i:

$$D_{i-2}^{-x} = D_{i-1}^{-x} = D_i^{-x} \tag{5.11}$$

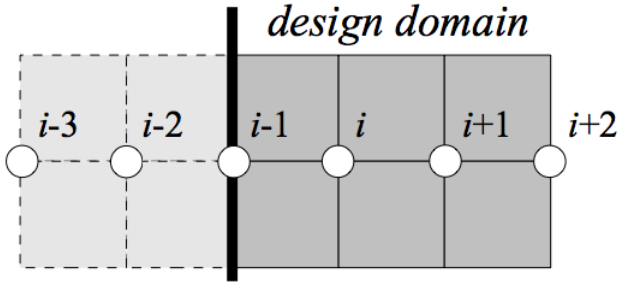

Figure 5-2: Left biased gradient approximation stencil

One final problem associated with the implemented upwind finite difference scheme is the gradient computation in the corners of the design domain, especially during the first iteration. The problem arises from the initialisation of the implicit function to be zero at the structural boundary. Thus, if the initial structure fills an entire corner of the design domain, Figure 5-3, the gradient component in each direction is zero for the corner node, producing a zero overall gradient (5.9). When this occurs, the implicit function value of the node diagonal to the corner node is used to estimate the gradient:

$$|\nabla \phi_c| = |\phi_c - \phi_a| \times \sqrt{2}/h \tag{5.12}$$

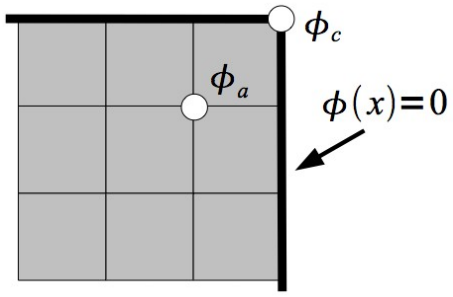

Figure 5-3: Corner node gradient approximation

5.2.4 Termination criterion

The necessary optimality condition for shape optimisation is that the velocity function is zero along the boundary (Allaire et al. 2004). This is equivalent to the shape sensitivity having a constant value along the boundary (M.Y. Wang et al. 2003). This condition can be used to formulate a termination criterion based on the integral of the velocity magnitude along the boundary:

$$\int_{\Gamma} |V_n| d\Gamma \leq \gamma \tag{5.13}$$

where γ is a small positive value. The small value is required because the fixed grid finite element analysis is likely to produce errors in sensitivity along the boundary, even when using a smoothing or least squares scheme. Furthermore, choosing an appropriate value of γ may require some experience and can be problem dependent. Therefore, a simpler heuristic termination criterion is employed, based on the maximum change in objective function over the previous ten iterations:

$$\Delta C^k = \frac{\left(C^m_{max} - C^m_{min}\right)}{\left(C^m_{max} + C^m_{min}\right)} \quad , \quad m \in [k-9, k] \tag{5.14}$$

where C^k is the objective computed at iteration k, C^m_{max} and C^m_{min} are the maximum and minimum objective values over the last 10 iterations and ΔC^k is a measure on the maximum relative change in objective function over the last 10 iterations. The optimisation process is stopped if $\Delta C^k < \gamma$.

5.3 Volume constraint

The volume constraint for the compliance problem, (1.1) is enforced by the Lagrange multiplier, λ when computing the velocity function (5.4). The value of λ should be computed at each iteration so that the solution remains feasible, or at least moves towards the feasible region. The bi-sectioning algorithm suggested by S. Y. Wang et al. (2007) computes a λ value that exactly satisfies the constraint, guaranteeing constraint satisfaction in the solution. A similar approach is introduced for this implementation. However, it was observed that the relationship between λ and volume change is often approximately linear. Therefore, the value of λ is efficiently calculated at each iteration using Newton's method.

5.3.1 Volume change computation

Volume change can be calculated by integrating the velocity function, (5.4) over the free boundary:

$$\Delta Vol(\lambda) = \Delta t \int_{\Gamma_0} V_n \, d\Gamma_0 = \Delta t \int_{\Gamma_0} \lambda - \tilde{E}_{ijkl}\, \varepsilon(u)_{ij}\, \varepsilon(u)_{kl} \, d\Gamma_0 \qquad (5.15)$$

where $\Delta Vol(\lambda)$ is the reduction in structure volume for a given λ value. A numerical approximation of this boundary integral can be constructed from the sensitivity values computed along the boundary. First, the length of boundary influenced by each discrete boundary velocity value is computed. Discrete boundary velocities are computed at intersections of the structure boundary and the fixed grid, Figure 5-4.

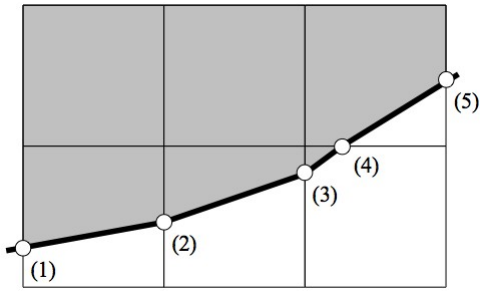

Figure 5-4: Boundary length approximation

Boundary length $|\Gamma|_i$ is approximated for each point, i by averaging the distance to neighbouring points along the boundary :

$$|\Gamma|_i = (d_1 + d_2)/2 \tag{5.16}$$

where d_1 and d_2 are distances from point i to its two neighbours:

$$\begin{aligned} d_1 &\approx [(y_i - y_{i-1})^2 + (x_i - x_{i-1})^2]^{1/2} \\ d_2 &\approx [(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2]^{1/2} \end{aligned} \tag{5.17}$$

The numerical approximation to (5.15) is then simply:

$$\Delta Vol \approx \Delta t \sum_{i=1}^{m} V_{n,i} |\Gamma|_i \tag{5.18}$$

where m is the number of discrete boundary points, $V_{n,i}$ is computed from (5.4) for a trial λ value and the time step is defined by (5.3). However, the calculation is complicated by the restriction that the structure cannot move beyond the design domain. Therefore, velocity values for discrete boundary points within one grid space, h of the design domain edge are checked for this restriction. If the restriction is violated then velocity values are modified to ensure the structure does not move beyond the design domain:

$$\tilde{V}_{n,i} = max\left(V_{n,i}, -d_\Gamma / \Delta t \right) \tag{5.19}$$

where $\tilde{V}_{n,i}$ is the modified velocity and d_Γ is the minimum distance from the point i to the design domain edge. However, modifying velocity values may alter the

absolute maximum value used to define the time step (5.3). Therefore, a short iterative process is required to obtain appropriately modified velocities and the correct time step. The full algorithm for estimating volume change for a trial λ value is then:

Step 1: Compute discrete boundary lengths (5.16), initial velocity values for a trial λ value (5.4) and initial Δt (5.3).

Step 2: For all boundary points within h of Ω edge, check structure does not move beyond the edge and modify velocity if necessary (5.19).

Step3: If $|V_n|_{max}$ has been altered in Step 2, then re-compute Δt and return to Step 2 without modified velocity values.

Step4: Estimate ΔVol using (5.18).

5.3.2 Newton's method

Using the numerical approximation of (5.15) detailed above, Newton's method can be employed to find the value of λ that produces a target volume change, ΔVol^t. First the volume change, ΔVol^0 for an initial estimate, λ^0 is computed. A good initial estimate for λ^0 was found to be the λ value computed for the previous level set iteration. If ΔVol^0 is within tolerance of ΔVol^t then no further iterations are required. Otherwise, a second estimate is constructed by comparing ΔVol^0 and ΔVol^t. If $\Delta Vol^0 < \Delta Vol^t$, then $\lambda^1 = 2\lambda^0$, otherwise, $\lambda^1 = 0.5\lambda^0$. Newton's method is then used to iteratively find a λ value that produces a volume change within 1% of the target volume:

$$\lambda^{m+1} = \frac{\Delta Vol^t - \Delta Vol^m}{\Delta Vol^m - \Delta Vol^{m-1}}\left(\lambda^m - \lambda^{m-1}\right) + \lambda^m \tag{5.20}$$

where m is an iteration in the Newton's method sub-loop.

The target volume is set to the difference between current and constraint values: $\Delta Vol^t = Vol^k - Vol^*$. However, if the volume is far from the constraint then the target may not be achievable under the CFL condition, thus an upper limit is defined in terms of the free boundary length:

$$|\Delta Vol^t|_{max} = 0.1\, h\, |\Gamma_0| \tag{5.21}$$

where $|\Gamma_0|$ can be computed by summing discrete values along the boundary (5.16). This limit is chosen from experience to ensure a reasonably smooth progression of the structure whose volume is far from the constraint.

5.4 Examples

The level set based optimisation method developed and implemented in this chapter is tested and validated using classic examples. All examples have material properties, E=1.0, v=0.3 and are discretized using unit sized square elements.

5.4.1 Cantilever beam

The first example is a cantilever beam of aspect ratio 2:1 with a single load on the right edge, Figure 5-5. The design domain is discretized using 160 × 80 elements and the volume constraint is set to 50% of the entire design domain. The effect of the termination criterion is investigated and the solution shown in Figure 5-6a is obtained after 64 iterations by setting $\gamma = 10^{-3}$. If a smaller criterion is used, $\gamma = 0.5\times10^{-3}$ then the solution shown in Figure 5-6b is obtained after 331 iterations. The final compliance values for the two solutions are 59.4×10^2 and 59.2×10^2 respectively. Both solutions have identical topology, but slightly differ in shape. However, both are in good agreement with results from another method (Takezawa et al. 2010).

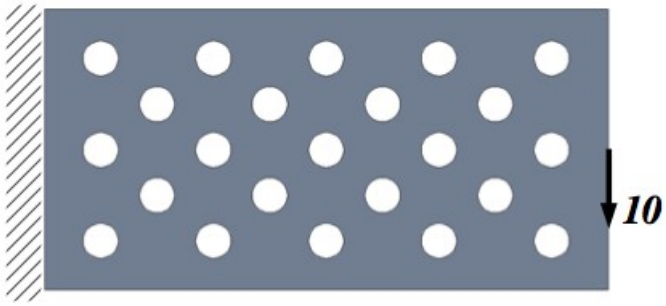

Figure 5-5: Cantilever beam, initial design and boundary conditions

For this example, significantly more iterations are required to meet a smaller, more rigorous termination criterion. However, the number of iterations required to meet the smaller criterion appears to be increased by oscillations in the compliance value after the larger criterion is met, Figure 5-7. These oscillations may be caused by small fluctuations in boundary sensitivity that only become significant near the optimum solution. For this example the fluctuations appear strong enough to artificially delay the smaller termination criterion being met. This highlights the difficulty in selecting an appropriate termination criterion.

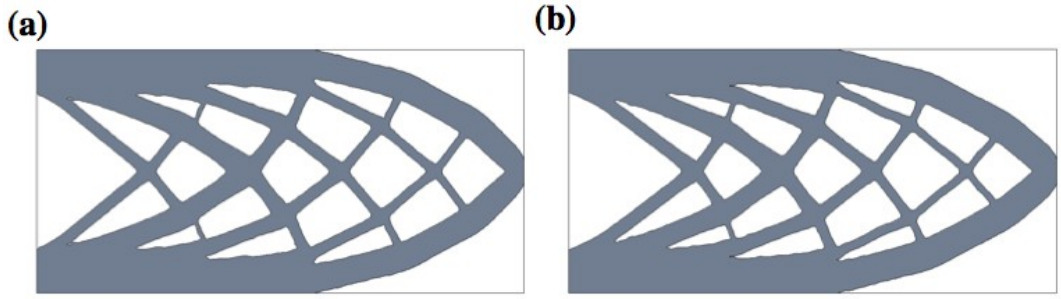

Figure 5-6: Cantilever beam solutions. a) $\gamma = 10^{-3}$, b) $\gamma = 0.5\times10^{-3}$.

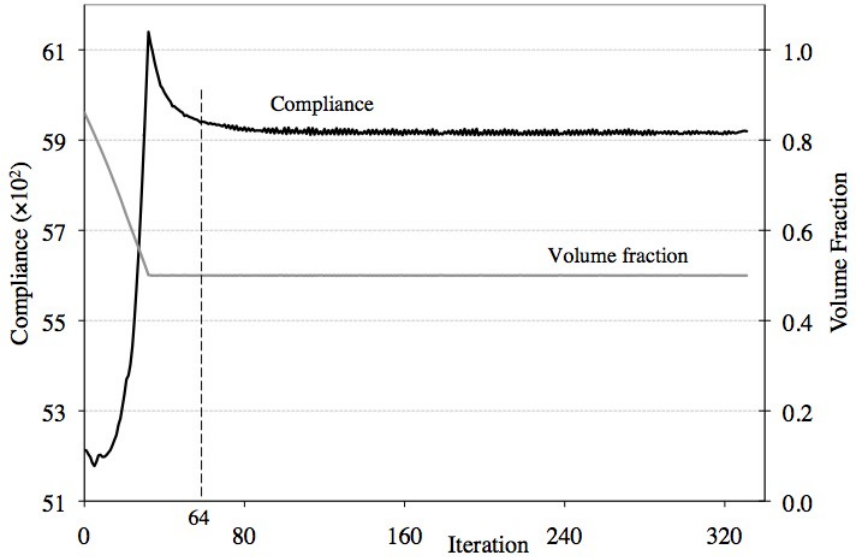

Figure 5-7: Cantilever beam, convergence history.

5.4.2 Michell structures

Michell type structures are widely used as benchmark examples for the minimum compliance topology optimisation problem. The structure with initial design and boundary conditions defined in Figure 5-8 is discretized by 160 × 80 elements and optimised for a volume constraint equal to 40% of the design domain. The solution obtained using the larger criterion, $\gamma = 10^{-3}$, agrees well with solutions from other methods (X. Wang et al. 2004; Mei & X. Wang 2004; S.Y. Wang & M.Y. Wang 2006a), although it contains some thin bars, Figure 5-9a, which disappear in the solution obtained using the smaller termination criterion, $\gamma = 0.5\times10^{-3}$, Figure 5-9b. The compliance values for these two solutions are 11.0×10^{2} and 10.9×10^{2}, for the larger and smaller criteria respectively. For this example, oscillations in compliance are not significant and do not delay the smaller criterion being met, Figure 5-10. The results from this example and the cantilever beam demonstrate that an appropriate universal termination criterion is difficult to define.

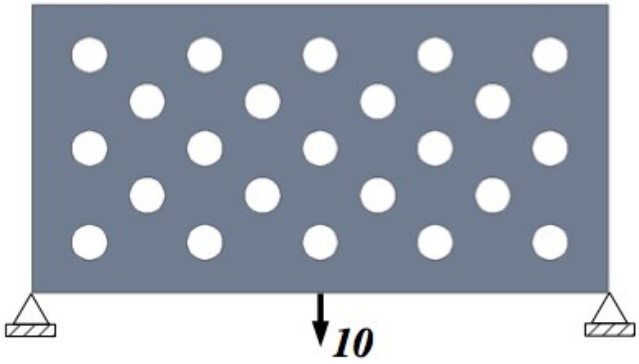

Figure 5-8: Michell structure, initial design and boundary conditions.

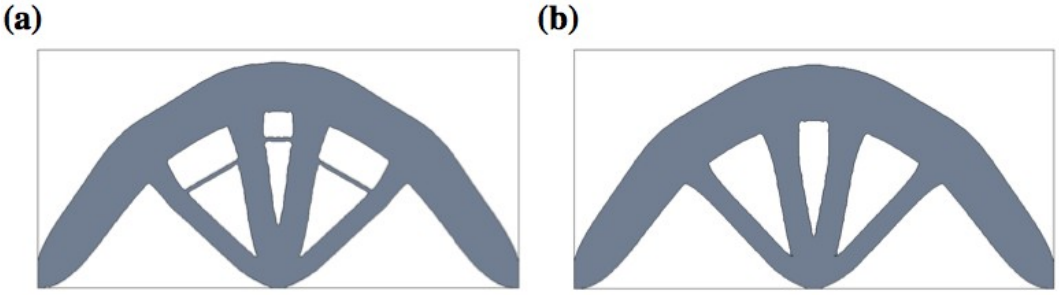

Figure 5-9: Michell structure solutions. a) 106 iterations, $\gamma = 10^{-3}$, b) 145 iterations, $\gamma = 0.5\times10^{-3}$.

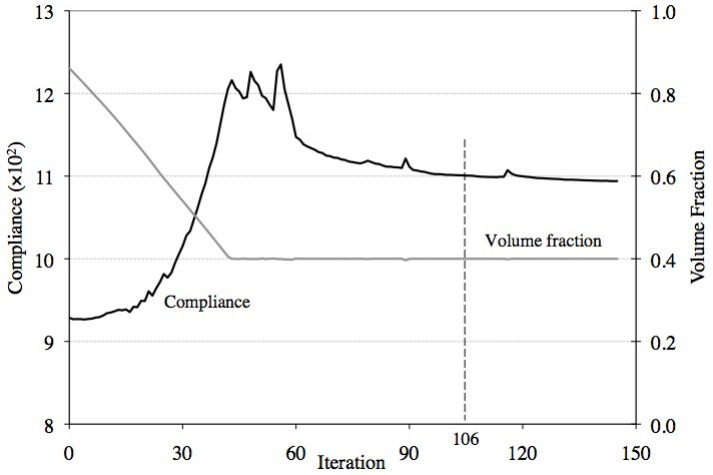

Figure 5-10: Michell structure convergence history.

A second Michell structure with different boundary conditions is now considered, Figure 5-11a. The structure is discretized using 160 × 80 elements, the volume constraint is set to 40% of the design domain and the smaller convergence criterion, $\gamma = 0.5\times10^{-3}$, is used. The solution contains features similar to those obtained from other methods (X. Wang et al. 2004; Norato et al. 2007; J. Luo et

al. 2008), Figure 5-11b and has a compliance value of 16.3×10^2, although the holes that appear inside the outer arch are usually not present in other solutions. However, if the optimisation is run without checking the termination criterion, then these holes reduce, Figure 5-11c, and eventually disappear, Figure 5-11d. This solution agrees more closely with those form other methods and has a lower compliance value of 15.8×10^2. This suggests that the additional holes in Figure 5-11b are not optimal, but exist in the solution due to early termination.

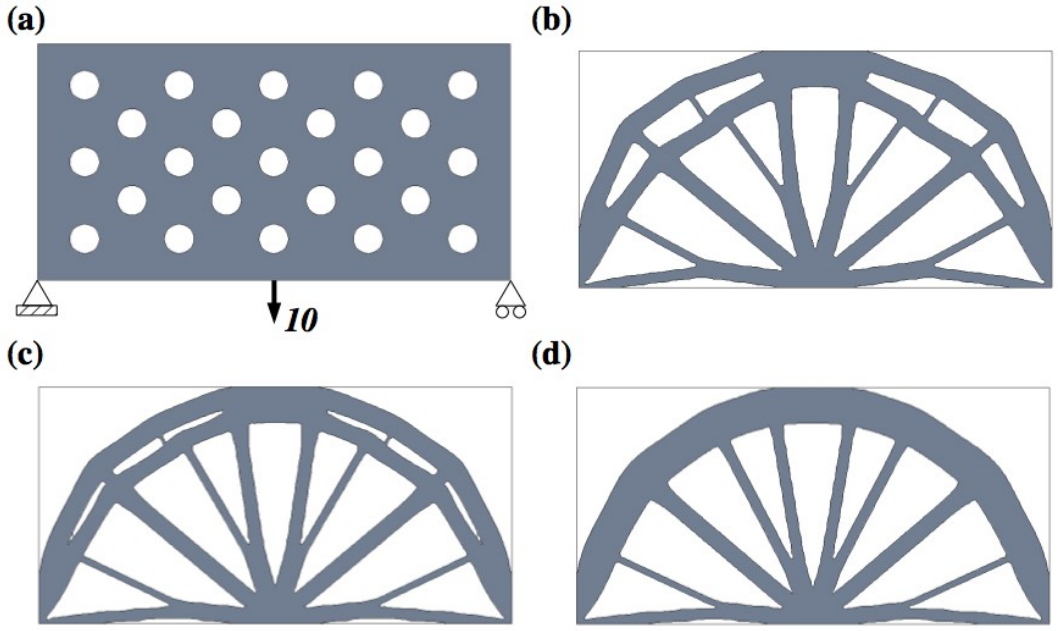

Figure 5-11: Second Michell structure. a) Initial design and boundary conditions, b) Solution after 80 iterations, c) 400 iterations, d) 700 iterations.

The second Michell structure example is optimised for an initial design with no initial holes, Figure 5-12a. The solution also contains no holes, Figure 5-12b and the compliance value is 28.3×10^2, which is significantly greater compared with the initial design with holes. This is a well known problem for the direct level set method, as there is no facility to insert new holes during optimisation.

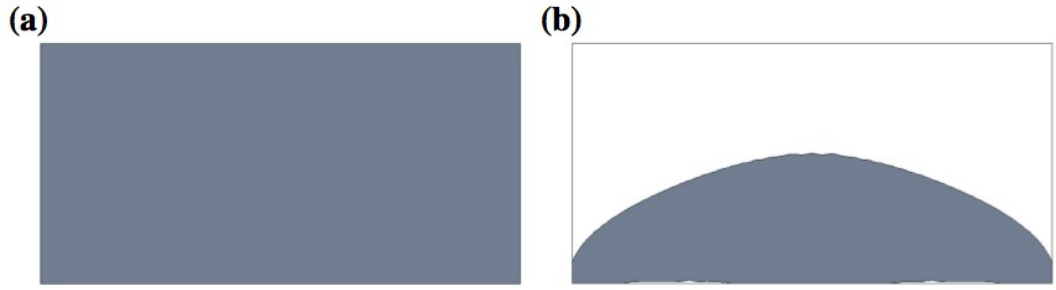

Figure 5-12: Second Michell structure. a) Initial design with no holes b) Solution after 317 iterations.

5.4.3 MBB beam

The MBB beam problem is also a popular benchmark example for structural topology optimisation (Bulman et al. 2001). Using symmetry conditions about the vertical axis, only the right half of the beam is considered, Figure 5-13. The volume constraint is set to 40% of the design domain and the termination criterion is $\gamma = 0.5\times10^{-3}$. The problem is solved using two different meshes.

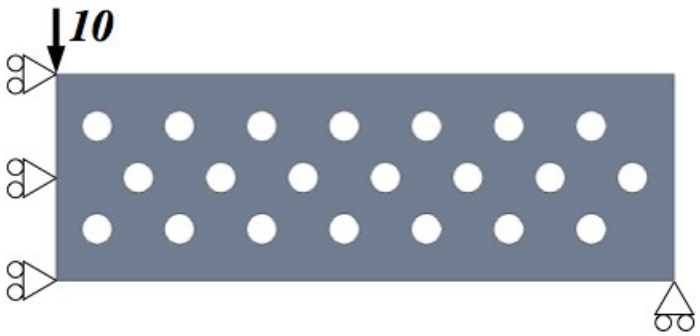

Figure 5-13: MBB beam, initial design and boundary conditions.

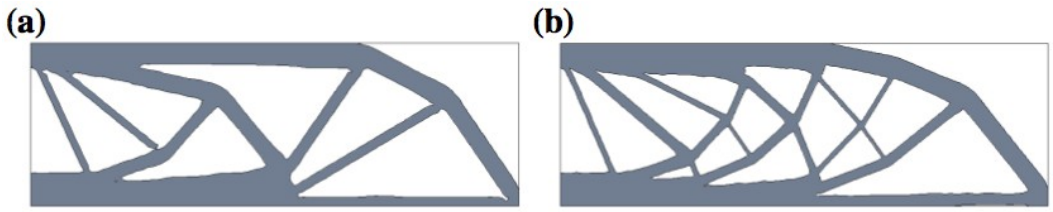

Figure 5-14: MBB beam. a) Solution for coarse mesh, b) Solution for fine mesh.

A coarse mesh of 120 × 40 elements produces a solution with compliance of 212.6×10^2, Figure 5-14a. The solution obtained using a finer mesh of 240 × 80 elements is similar, but contains additional thin bars, Figure 5-14b. These additional bars help slightly reduce the compliance to 211.6×10^2. The thin bars are difficult to represent in the coarse mesh using a discretized implicit function because their thickness is less than the grid spacing. However, both solutions agree well with those obtained using other methods (Bendsøe & Sigmund 2004; Mei & X. Wang 2004; Norato et al. 2007).

The above example shows that the selection of mesh density has an affect on the optimal solution obtained. A finer mesh allows for finer geometrical details to be represented, which can lead to better designs with lower compliance values. A further potential benefit of using a finer mesh is more accurate analysis leading to a more accurate sensitivity computation. However, when using the AFG method convergence of displacement and stress with increasing mesh density may not be monotonic, although using the least squares method helps alleviate this effect (Wei et al. 2010; Garcia-Ruiz & Steven 1999). The disadvantage of using a finer mesh is increased computational cost. For the level set method this occurs is two ways, increased cost of performing the FEA and slower convergence as boundary movement is restricted by the CFL condition (5.3).

Selecting mesh density for level set based optimization is a trade-off between computational cost and potentially obtaining a better, more detailed solution. Mesh densities are selected throughout this thesis to provide sufficient detail in the solutions to allow for meaningful comparisons at a reasonable computational cost. This is achieved in an ad hoc fashion by solving problems using a corse mesh then increasing the mesh density until reasonable solutions were obtained.

5.4.4 General performance of the method

The examples converged reasonably smoothly towards an optimum solution, which demonstrates that the implemented level set optimisation method is numerically stable. The signed distance property of the implicit function is maintained well within the narrow band region and gradient estimation is accurate and robust throughout the optimisation. Sensitivity computation using the AFG method and proposed weighted least squares scheme (Chapter 3) appears sufficient to progress the design reasonably smoothly towards an optimum solution. However, small spurious fluctuations in boundary sensitivity arising from the discretisation can prevent or delay the termination criterion being met when the design is near the optimum. However, setting a larger criterion can result in early termination. Thus, defining an appropriate and universal termination criterion is difficult.

The volume constraint was maintained or made more feasible using the proposed numerical estimate for volume change and Newton's method. The approach was reasonably efficient for all examples, as often only 3 or 4 iterations of Newton's method were required to compute the necessary λ value.

The solutions obtained for the benchmark examples all agree well with solutions from other methods, thus validating the implementation. However, some solutions are dependent on the initial design. This is mainly because the implemented method does not have the ability to create additional holes during optimisation. This is a well known problem for level set based optimisation methods and is discussed further in the next chapter.

5.5 Conclusions

A stable implementation of a direct level set optimisation method was presented and validated against classic benchmark examples. The results showed that the implemented method obtained solutions consistent with those from other methods. Although, solutions can be dependent on the initial design. The AFG method with the weighted least scheme proposed in Chapter 3 provided shape sensitivities with sufficient accuracy and numerical instability was not observed during optimisation. However, oscillations did occur near the optimum, which could artificially delay convergence and made it difficult to define a universal termination criterion. Finally, the Newton's method approach for computing the volume constraint Lagrange multiplier was effective and efficient.

Chapter 6

A hole insertion method for level set based optimisation

6.1 Introduction

Solutions obtained using the level set optimisation method presented in the previous chapter can be dependent on the initial design. This limitation is also observed in similar methods and is largely because there is no mechanism to create new holes during the optimisation (Allaire et al. 2004; Mei & Wang 2004; K. Park & Youn 2008). In this sense the direct level set method is similar to a shape optimisation method, but with the capacity to automatically merge holes through the implicit boundary representation. This enables some change in topology during the optimisation, but new holes cannot be created.

Various methods have been proposed to facilitate new hole creation during optimisation using the level set method. These are reviewed in Chapter 2.3. In general, methods that employ topological derivatives do not formulate a clear link between boundary movement and hole insertion and solutions obtained from alternate methods can be sensitive to various parameters.

A new approach is created in this work to enable hole insertion for the direct level set optimisation method detailed in the previous chapter. The method is presented in a general context, but is applied to solve the minimisation of compliance problem (1.1).

6.2 Hole creation method

It has been observed that new holes can emerge naturally in 3-d problems when two zero level set surfaces cross without breaking the connectivity of the shape or void (Allaire et al. 2004). It is proposed to exploit this phenomenon to facilitate natural hole creation in 2-d problems. To mimic the hole insertion mechanism that occurs in 3-d problems, a secondary implicit level set function is introduced, $\overline{\phi}(x)$, to represent a pseudo third dimension for the 2-d continuum. The pseudo third dimension acts as a fictitious thickness for the 2-d structure. However, it is assumed that the thickness is sufficiently small compared to the dimensions of the 2-d structure, allowing thickness effects to be ignored. The secondary implicit level set function is initialised to an artificial height, $\overline{h}$ above the structure domain:

$$\overline{\Phi}^{0}(x)=\begin{cases}+\overline{h}\ , & x\in\Omega_S\\ -\overline{h}\ , & x\notin\Omega_S\end{cases} \tag{6.1}$$

This values of the initialisation are also used to define the upper and lower bounds of the secondary implicit function: $-\overline{h} \le \overline{\phi}(x) \le \overline{h}$. Update of the secondary implicit function during each iteration of the optimisation is performed using:

$$\overline{\Phi}_i^{k+1}=\overline{\Phi}_i^{k}-\Delta t V_{n,i} \tag{6.2}$$

where velocity values are computed at internal nodes from sensitivity values in the same manner as boundary velocities. For the minimisation of compliance problem (1.1) velocities can be defined from sensitivities as:

$$V_{n,i} = \bar{\lambda} - \varsigma_i \tag{6.3}$$

where ς_i are shape sensitivity values computed at internal nodes using (2.13). Using shape sensitivities computed at points inside a structure has also been used to create new holes in other methods (Belytschko et al. 2003; K. Park & Youn 2008). This approach can be justified by the clear link between shape and topological gradients, as demonstrated by Céa et al. (2000).

A new hole is created when $\bar{\phi}(x)$ becomes negative within the region of Ω_S and the new hole is added to the primary level set function by simply copying $\bar{\phi}(x)$ onto $\phi(x)$ within Ω_S. The progression of the secondary implicit function is linked to the primary one by using a common value for Δt in (5.2) and (6.2) and a consistent velocity definition by setting $\bar{\lambda} = \lambda$ in (6.3). The main advantage of this approach is that holes are allowed to emerge naturally during the optimisation process.

6.3 Implementation of the method

In practice some care is required when utilising the secondary implicit function as a device for new hole insertion. Firstly the choice of the initial artificial height, $\bar{h}$ in (6.1) affects the ease and frequency that holes can emerge. A larger value of $\bar{h}$ represents a thicker structure causing holes to emerge more gradually, whereas a smaller $\bar{h}$ value allows holes to emerge more frequently.

In the narrow band region, the primary implicit function is updated using extension velocities derived from boundary movement. Therefore, the implicit function value can be updated by either the primary or secondary implicit function within the narrow band. This choice of update is removed by limiting hole insertion to the part of Ω_S that is not part of the narrow band region. This approach has the additional effect that the narrow band width acts like a minimum member width where new holes cannot emerge.

The primary implicit level set function is unlikely to remain a signed distance function when a new hole is created, as values are simply copied from the secondary function. Thus, the primary implicit function may become too flat or steep around newly created holes, which can adversely affect the stability of the method. This is avoided by reinitialising the primary implicit function to be a signed distance function after a new hole is inserted, using the method discussed in Section 5.2.2. It is also beneficial to reinitialise the secondary implicit function using (6.1), whenever the primary function is reinitialised. The implicit function is also reinitialised after hole insertion for methods employing topological derivatives (Allaire et al. 2005; Zhuang et al. 2007; M. Kim et al. 2009).

6.3.1 New hole volume limit

The velocity function used to update the secondary implicit function, (6.2) is based on the derivative of the objective function. Therefore, the emergence of new holes through secondary implicit function update is similar to the gradient descent method. Hence, a move limit is introduced to prevent large steps occurring during optimisation. A limit, $\overline{Vol}$ is applied to the maximum volume of material removed from the structure when new holes are inserted. If this limit is exceeded, then the value of $\bar{\lambda}$ is re-computed so that the limit is satisfied.

The volume of material removed due to hole insertion can be numerically estimated for a trial value of $\bar{\lambda}$. First, a temporary updated secondary implicit function, $\overline{\phi}_t$ is computed using (6.2), (6.3) and the value of $\bar{\lambda}$. If no value of $\overline{\phi}_t$ within the structure becomes negative then no new holes are created and $\overline{\phi}_t$ becomes the updated secondary implicit function. Otherwise, the new hole volume is computed by summing volume estimates from each node with a negative $\overline{\phi}_t$ value that lies within the structure, but outside the narrow band, using neighbouring values, Figure 6-1.

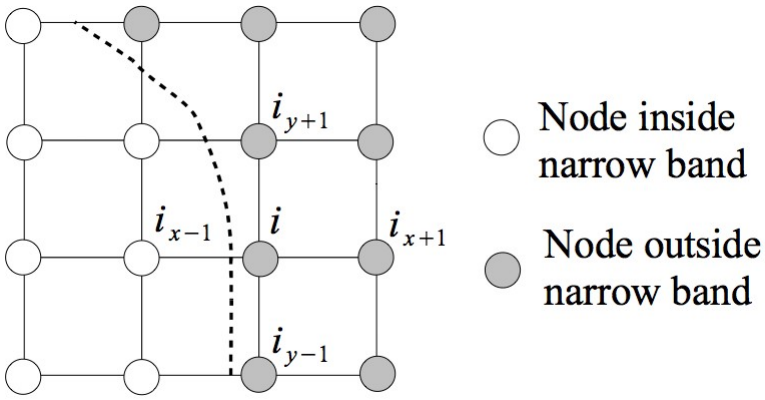

Figure 6-1: Hole volume estimate for a node near the narrow band region.

To improve the stability of the method, it was found necessary to prevent very small holes being created. A small hole is identified if the four neighbouring $\overline{\phi}_t$ values to a node with negative $\overline{\phi}_t$ are all positive. If this occurs then small hole creation is prevented at the node by assigning a small positive $\overline{\phi}_t$ value. Once this check has been performed, new hole volume around a node, i is estimated using:

$$Vol_i = \frac{h^2}{4} \sum_j \Phi_i(\overline{\Phi}_{t,j}) \ , \ j \in [i_{x-1}, i_{x+1}, i_{y-1}, i_{y+1}] \tag{6.4}$$

where $\Phi_i(\overline{\phi}_{tj})$ is defined by:

$$\Phi_i(\overline{\Phi}_{t,j}) = \begin{cases} \overline{\Phi}_{t,i} / (\overline{\Phi}_{t,i} - \overline{\Phi}_{t,j}) \ , \ if \ \overline{\Phi}_{t,j} \geq 0 \\ 1 \qquad\qquad\qquad\qquad , if \ \overline{\Phi}_{t,j} < 0 \end{cases} \tag{6.5}$$

However, the secondary function is not copied onto the narrow band region of the primary function. Thus, the value for $\overline{\phi}_j$ used in (6.5) for nodes inside the narrow band is the current primary function value.

If the new hole volume estimated by summing values computed using (6.4) for $\overline{\phi}_t$ with $\overline{\lambda} = \lambda$ is greater than the limit, then $\overline{\lambda}$ is modified to meet the limit using an iterative approach. The new hole volume is a monotonic function of $\overline{\lambda}$ and Newton's method is used to compute the modified $\overline{\lambda}$ value. The iterative process is terminated when a new hole volume estimate is within 1% of the limit value.

If $\bar{\lambda}$ is modified to meet the new hole volume limit, so that $\bar{\lambda} \neq \lambda$, then the link between primary and secondary function update is invalidated. Thus, primary function update by boundary propagation (5.2) is not performed if $\bar{\lambda} \neq \lambda$. Normally, if $\bar{\lambda} = \lambda$, then both hole insertion and boundary propagation can be performed during the same iteration. The complete level set optimisation algorithm, with the new hole creation method, is illustrated in Figure 6-2.

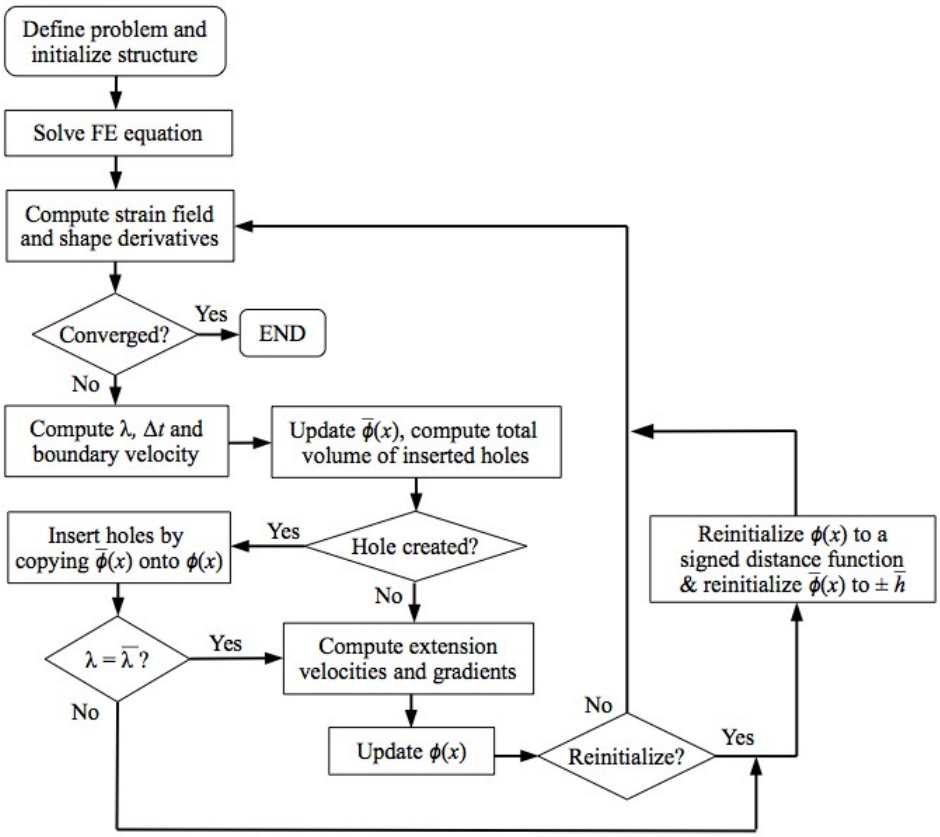

Figure 6-2: Level set optimisation algorithm with new hole creation method.

6.4 Parameter investigation

Although the proposed method includes a meaningful link between hole creation and boundary propagation, it is not completely free of parameters that can potentially affect the solution obtained. These parameters are: narrow band width, ω new hole volume limit, $\overline{Vol}$ and initial artificial height, $\overline{h}$. The effect of parameter choice can be partially alleviated by defining them in terms of other parameters used in the method. Band width and artificial height are defined in terms of the grid spacing, h and hole volume limit is defined as a percentage of the current structure volume. The effects of these parameters are investigated using numerical examples to identify suitable values that generally produce good results.

6.4.1 Cantilever beam

The cantilever beam example from Chapter 5 (Figure 5-5) is used to investigate the hole insertion method parameters. The initial beam design contains no holes and the design domain is completely filled with material. A range of values are chosen for each parameter: $\overline{h}$ = 0.5h, h, 2h; $\overline{Vol}$ = 0.5%, 1%, 2% of Ω_S; ω = 4h, 6h. The volume constraint is set to 50% of the design domain and the minimisation of compliance problem is solved for each combination of parameter values. This produces a range of solutions, which are summarised in Figure 6-3 (ω = 4h) and Figure 6-4 (ω = 6h). The termination criterion, γ=0.5×10^{-3} (5.14) was used for all problems.

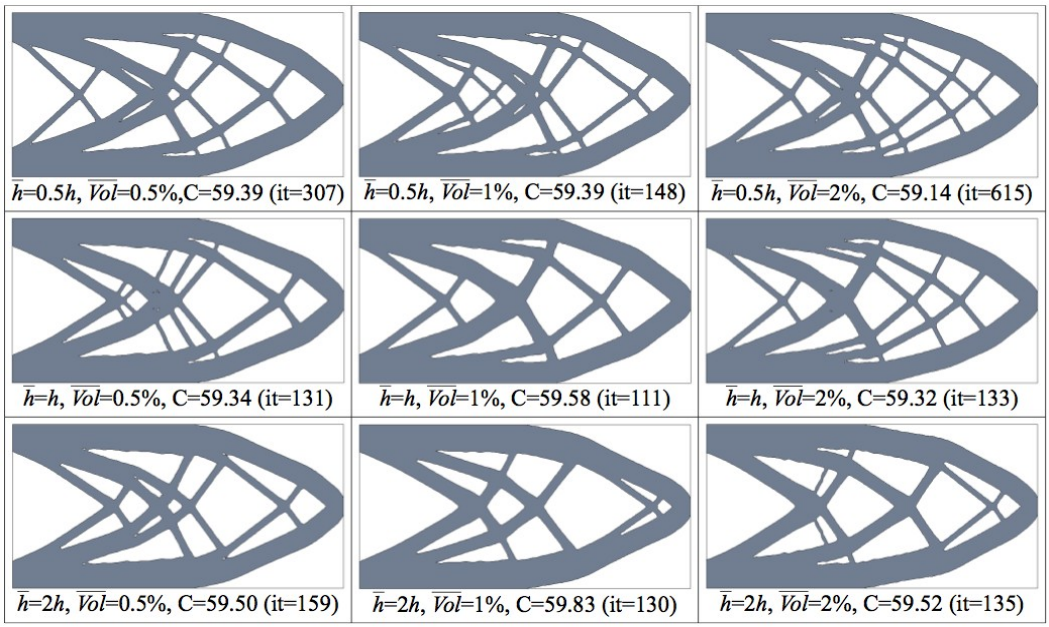

Figure 6-3: Cantilever beam solutions for band width, $\omega = 4h$. Compliance values, (C, $\times 10^2$) are for final solution at the iteration (it) shown.

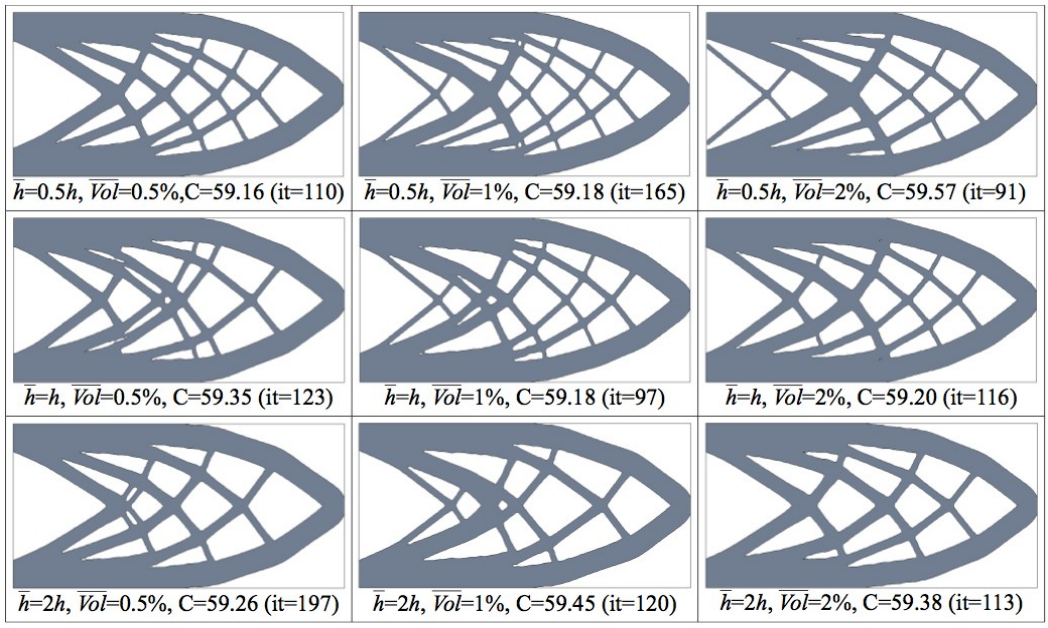

Figure 6-4: Cantilever beam solutions for band width, $\omega = 6h$. Compliance values (C, $\times 10^2$) are for final solution at the iteration (it) shown.

The results of the investigation show that different solutions are obtained when using different parameter values. However, all solutions are reasonably similar and final compliance values are all within around 1%. This suggests that solutions are not significantly affected by the chosen parameters. However, some generalisations on the effect of each parameter may be drawn from this example. In general the larger band width, $\omega = 6h$ produces solutions with more consistent shape and topology that converge in fewer iterations compared with $\omega = 4h$. Lower values of the artificial height, $\overline{h}$ allow new holes to emerge more frequently during optimisation and, as a consequence, solutions obtained with lower values tend to possess more holes than solutions obtained using higher $\overline{h}$ values. Also, smaller values for the new hole volume limit, $\overline{Vol}$ were active more often than larger values. Furthermore, the largest limit considered, $\overline{Vol}$ =2% was often only active once or twice during optimisation and not active at all for $\omega = 6h$, $\overline{h} = 1.0$. Therefore, a larger limit on new hole volume is less likely to disrupt the optimisation by having to modify the $\overline{\lambda}$ value, which prevents update by boundary propagation. However, a limit is still required to prevent too much material being removed in a single iteration.

6.4.2 Further investigation

The above investigation was repeated for the MBB beam, discretized with 120×40 elements, (Figure 5-14) and the Michell structure (Figure 5-11) examples and similar observations were made. Therefore, the results of these additional investigations are omitted for brevity. The overall results suggest that using a band width, $\omega = 6h$, artificial height, $\overline{h} = h$ and volume limit, $\overline{Vol}$ = 2% can produce good solutions in a few iterations for a range of examples. For demonstration, the solutions for the MBB beam (Figure 6-5a) and Michell structure (Figure 6-6a) using these parameters are shown, along with their convergence histories (Figure 6-5b, Figure 6-6b).

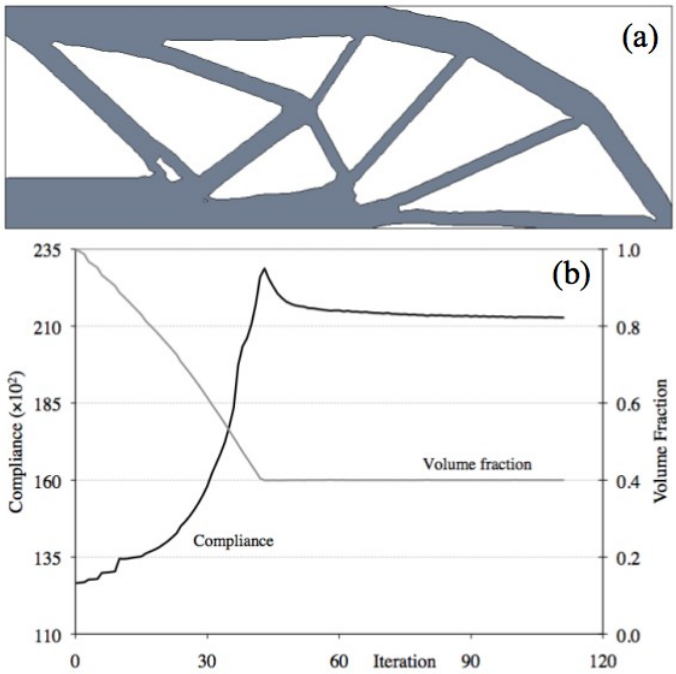

Figure 6-5: MBB beam. a) Solution at iteration 111, b) Convergence of compliance and volume constraint.

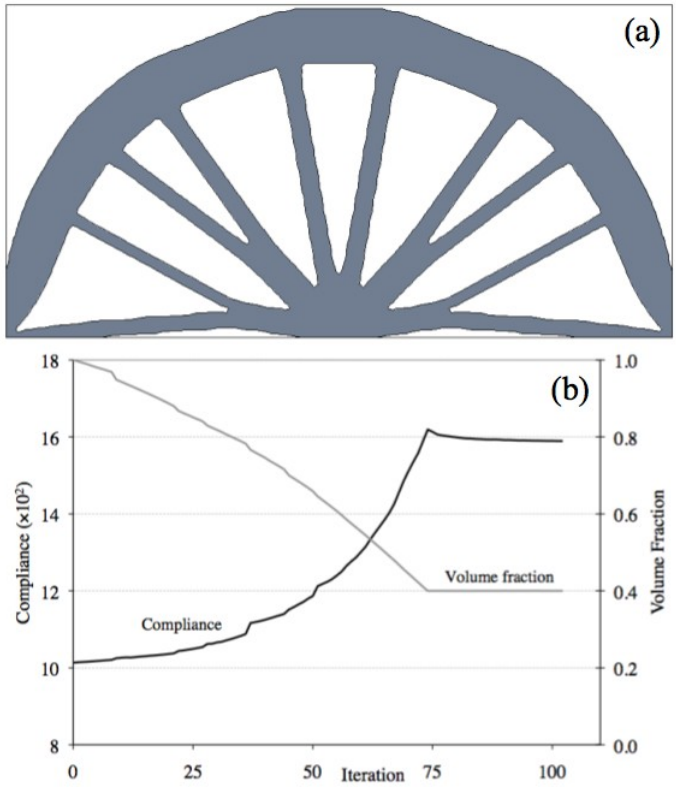

Figure 6-6: Michell structure. a) Solution at iteration 102, b) Convergence of compliance and volume constraint.

The investigation also allowed for some general observations on the performance of the overall hole insertion algorithm. Firstly, most examples converge reasonably smoothly to the solution, suggesting the link between hole insertion and boundary propagation is valid, as significant oscillations or discontinuities in compliance do not often occur. However, smoother convergence is more evident when using the larger band width, $\omega = 6h$. Most new holes are inserted during the initial stages of the optimisation, before the volume constraint is reached. After the constraint is reached, often only a few iterations are required to obtain the final solution. This suggests that new holes are created in useful locations and are retained in the final solution.

6.5 Conclusions

A new hole insertion method was created for a 2-d direct level set based optimisation method. Hole creation was linked to front propagation through update of a pseudo third dimension represented by a secondary implicit function. The proposed method allowed for simultaneous and unified optimisation of both shape and topology through a unified update of the primary and secondary implicit level set functions.

An extensive parameter investigation showed that using a larger narrow band width produced more consistent solutions in fewer iterations. Using a smaller artificial height for the pseudo third dimension allowed holes to emerge more frequently and consequently, solutions tended to posses more holes. Also, a smaller limit on new hole volume was often more active than larger limits. Thus, larger limits were less likely to disrupt the optimisation by preventing update by boundary propagation. However, a limit was still required to prevent too much material being removed in a single iteration. Although the choice of parameters effected solutions, final compliance values were not significantly different for the solutions obtained using the range of parameters considered.

The investigation also demonstrated the performance of the overall method. Most examples converged reasonably smoothly to the solution, suggesting the link between hole insertion and boundary propagation is valid, as significant oscillations or discontinuities in compliance did not often occur. Most new holes were inserted early on, before the volume constraint was satisfied and often relatively few iterations were required to obtain the final solution. This suggests that new holes emerged in useful locations that were retained in final solutions. Overall, the new hole insertion method was able to smoothly obtain good solutions for a range of examples in relatively few iterations and was not significantly sensitive to the choice parameters.

Chapter 7

Minimisation of expected compliance under loading uncertainty

7.1 Introduction

Real operating environments are full of uncertainties that should be included in structural optimisation methods to produce robust structures. Robust optimisation uses a probabilistic objective function to account for uncertainty. There are various uncertainties that can affect the robustness of a structure, including: loads, material properties and geometry. In this work uncertainties in loading conditions are introduced into the minimisation of compliance problem (1.1) using the robust optimisation approach. Both loading magnitude and direction uncertainties are considered and assumed to be statistically independent and normally distributed.

This chapter considers the minimisation of expected compliance robust optimisation problem:

$$\begin{aligned} &\text{Minimise: } E[C(u,f)] = \int_f C(u,f)P(f)df \\ &\text{Subject to: } \int_{\Omega_s} d\,\Omega_S \le Vol^* \\ &\qquad \int_{\Omega_s} \tilde{E}_{ijkl}\,\varepsilon(u)_{ij}\,\varepsilon(v)_{kl}\,d\,\Omega_S = \int_{\Gamma_s} fv\,d\,\Gamma_S \end{aligned} \tag{7.1}$$

where $E[C]$ is the expected value of compliance and $P(f)$ is the probability density function of the uncertain loading conditions.

Loading uncertainties are widely studied, although the level of uncertainty is often limited to loading magnitude. Robust optimisation methods often use discretization techniques to approximate uncertain loads described using continuous probability density functions. This transforms the optimisation problem into a multiple load case one, which can be easily solved (Bendsøe & Sigmund 2004; Allaire & Jouve 2005). In order to achieve a level of accuracy when discretizing continuous probability functions, increasing the number of sampling points or a higher order quadrature rule is required. This consequently increases the number of load cases that must be considered. Thus, in the presence of a large number of uncertain loads, the computational cost associated with the discretization approach can quickly become prohibitive (Evgrafov & Patriksson 2003; Calafiore & Dabbene 2008).

This chapter employs an analytical approach to produce an efficient method for considering loading magnitude and directional uncertainties when solving the minimisation of expected compliance problem (7.1). The proposed approach alleviates the computational burden of discretization techniques when there are a large number of uncertain loads and a high degree of accuracy is required.

7.2 Loading magnitude uncertainty

First, only loading magnitude uncertainty is considered. It is assumed that the static equilibrium equation in (7.1) is solved discretely using some form of FEA:

$$\{f\}=[K]\{u\} \tag{7.2}$$

where $\{f\}$ is a vector of nodal loads, $\{u\}$ the vector of nodal displacements and $[K]$ is a symmetric stiffness matrix of size $m \times m$, where m is the number of unconstrained degrees of freedom. Using (7.2) a discrete compliance function can be written as:

$$C(f)=\{f\}^T[K]^{-1}\{f\}=\sum_{i,j=1}^{m}\kappa_{i,j}f_i f_j \tag{7.3}$$

where $\kappa_{i,j}$ is an entry in the inverse stiffness matrix, $[K]^{-1}$. If magnitude uncertainties are statistically independent, expected compliance is:

$$E[C(f)]=\int_{f_n}\cdots\int_{f_1}\left(\sum_{i,j=1}^{m}\kappa_{i,j}f_i f_j\right)\prod_{i=1}^{n}P(f_i)\,df_1\cdots df_n \tag{7.4}$$

where $P(f_i)$ is the probability density function and n is the number of loads with uncertain magnitude. It is assumed that the degrees of freedom are numbered so that the first n correspond to the uncertain loads, $n \le m$. The integral in (7.4) is evaluated over the first uncertain load, f_1 using integration by parts three times:

$$\begin{aligned}E[C]=\int_{f_n}\cdots\int_{f_2}\Bigg[&\left(\kappa_{1,1}f_1^2+2f_1\sum_{i=2}^{m}\kappa_{1,i}f_i+\sum_{i,j=2}^{m}\kappa_{i,j}f_i f_j\right)\int P(f_1)df_1\\&-2\left(\kappa_{1,1}f_1+\sum_{i=2}^{m}\kappa_{1,i}f_i\right)\int\int P(f_1)df_1\\&+2\kappa_{1,1}\int\int\int P(f_1)df_1\Bigg]\prod_{i=1}^{n}P(f_i)\,df_2\cdots df_n\end{aligned} \tag{7.5}$$

If $P(f_i)$ for each uncertain magnitude is a normal distribution with mean μ_i and variance σ_i^2, (7.5) is evaluated between limits $\mu_1 \pm \zeta$, then letting $\zeta \to \infty$ yields:

$$E[C] = \int_{f_n} \cdots \int_{f_1} \left[(\mu_1^2 + \sigma_1^2)\kappa_{1,1} + 2\mu_1 \sum_{i=2}^{m} \kappa_{1,i} f_i + \sum_{i,j=2}^{m} \kappa_{i,j} f_i f_j \right] \prod_{i=1}^{n} P(f_i) df_1 \cdots df_n \tag{7.6}$$

For further details of this integral evaluation see Appendix 1.2. Repeating the integration process for each uncertain loading magnitude, the expected compliance can be simplified to:

$$E[C] = \sum_{i,j=1}^{m} \kappa_{i,j} \mu_i \mu_j + \sum_{i=1}^{n} \kappa_{i,i} \sigma_i^2 \tag{7.7}$$

The analytical derivation (7.7) reveals that expected compliance can be evaluated by summing values from n+1 deterministic load cases, where n is the number of loads with non-zero variance. The first load case is the simultaneous application of mean loads and the subsequent n cases correspond to a single load equal to σ_i applied at the location of the uncertain load. Therefore, the robust optimisation problem of (7.1) can be transformed into a multiple load case problem that can be solved by most existing topology optimisation methods (Bendsøe & Sigmund 2004; Allaire & Jouve 2005):

$$\begin{aligned} &\text{Minimise: } E[C(f)] = C(\mu) + \sum_{i=1}^{n} C(\sigma_i) \\ &\text{Subject to: } \int_{\Gamma} d\Gamma \le Vol^* \end{aligned} \tag{7.8}$$

where $C(\mu)$ is the compliance computed from mean loading conditions and $C(\sigma_i)$ is the compliance for a single load of magnitude σ_i. This formulation offers an efficient and exact method for evaluating expected compliance using the minimum number of deterministic load cases where loads have uncertainty in magnitude described by normal distributions.

7.3 Loading direction uncertainty

The analytical approach is now employed to develop an efficient method for calculating expected compliance in the presence of loading direction uncertainty. Directional uncertainty of a load in two-dimensional space is defined here in terms of its angle of application θ, where θ is a counter-clockwise angle from the x-axis. Expected compliance of a structure under n statistically independent loads with directional uncertainty can be written as:

$$E[C(f,\theta)] = \int_{\theta_n} \cdots \int_{\theta_1} C(f,\theta) \prod_{i=1}^{n} P(\theta_i)\, d\theta_1 \cdots d\theta_n \tag{7.9}$$

where $P(\theta)$ is a probability density function for directional uncertainty. It is assumed that directional uncertainty is normally distributed with mean direction $\mu_{\theta,i}$ and standard deviation $\sigma_{\theta,i}$. A single load with magnitude f, but applied in an arbitrary direction, can be written in terms of two orthogonal loads. For simplicity it is assumed one load is defined in the horizontal x-direction and the other in the vertical y-direction:

$$\begin{aligned} f_x(f,\theta) &= f\cos(\theta) \\ f_y(f,\theta) &= f\sin(\theta) \end{aligned} \tag{7.10}$$

The load vector $\{f\}$ is constructed such that odd entries correspond to horizontal loads and even entries to vertical loads. As only linear elastic structures are considered in this work, the stiffness matrix $[K]$ and its inverse are symmetrical. Substituting the orthogonal loads (7.10) into the discrete form of compliance (7.3) results in:

$$\begin{aligned} C(f,\theta) = \sum_{i,j=1}^{m/2} f_i f_j \big(& \kappa_{ix,jx}\cos(\theta_i)\cos(\theta_j) + \kappa_{iy,jy}\sin(\theta_i)\sin(\theta_j) \\ & + 2\,\kappa_{ix,jy}\cos(\theta_i)\sin(\theta_j) \big) \end{aligned} \tag{7.11}$$

where $ix = 2i\text{-}1$ and $iy = 2i$. Note that "ix" and "iy" are not products of i and x or y.

Substituting (7.11) into (7.9), the integral for expected compliance can be evaluated by utilising the complex exponential forms of the trigonometric functions. Taking the integration limits of each uncertain variable to be $\theta_i = \mu_{\theta,i} \pm \pi$, which integrates over the full revolution of 2π. The result of the integral for expected compliance involves error functions in the form: erf $[\pi / (\sigma_{\theta,i} \sqrt{2})]$. However, if at least three standard deviations either side of the mean direction are contained within the full revolution, $3\sigma_{\theta,i} \leq \pi$, then the error function evaluates to at least 0.9973, which can be approximated as one. Further details of the integration are shown in Appendix 1.2. The integral simplifies to:

$$\begin{aligned} E[C] = 1/2 \sum_{i=1}^{m/2} \Big[& \kappa_{ix,ix} (1 + \exp(-2\sigma_{\theta i}^2) \cos(2\mu_{\theta i})) \\ & + \kappa_{iy,iy} (1 - \exp(-2\sigma_{\theta i}^2) \cos(2\mu_{\theta i})) + \kappa_{ix,iy} (\exp(-2\sigma_{\theta i}^2) \sin(2\mu_{\theta i})) \Big] \\ + \sum_{i,j=1:\, j \neq i}^{m/2} & \Big(f_i f_j \exp(-\sigma_{\theta i}^2/2) \exp(-\sigma_{\theta j}^2/2) \Big) \Big[\kappa_{ix,jx} \cos(\mu_{\theta i}) \cos(\mu_{\theta j}) \\ & + \kappa_{iy,jy} \sin(\mu_{\theta i}) \sin(\mu_{\theta j}) + 2\kappa_{ix,jy} \cos(\mu_{\theta i}) \sin(\mu_{\theta j}) \Big] \end{aligned} \tag{7.12}$$

A number of load cases are now defined, whose compliance values sum to the expected value derived in (7.12). This is achieved by subtracting the compliance values for each load case from the expected compliance value. The first load case is defined as the mean one, where each entry in the load vector is multiplied by an exponential function of the variance:

$$C_1(\bar{f}, \mu_\theta) = \{\bar{f}\}^T [K]^{-1} \{\bar{f}\} \tag{7.13}$$

$$\bar{f}_i = f_i \exp(-\sigma_{\theta i}^2/2) \tag{7.14}$$

The compliance for the first load case is computed using (7.11):

$$\begin{aligned} C_1 = \sum_{i,j=1}^{m/2} & \Big[f_i f_j \exp(-\sigma_{\theta i}^2/2) \exp(-\sigma_{\theta j}^2/2) \Big] \Big[\kappa_{ix,jx} \cos(\mu_{\theta i}) \cos(\mu_{\theta j}) \\ & + \kappa_{iy,jy} \sin(\mu_{\theta i}) \sin(\mu_{\theta j}) + 2\kappa_{ix,jy} \cos(\mu_{\theta i}) \sin(\mu_{\theta j}) \Big] \end{aligned} \tag{7.15}$$

The compliance from the first load case (7.15) is now subtracted from the expected compliance (7.12) leaving the compliance value to be computed by the remaining load cases. After some manipulation the subtraction equates to:

$$E[C]-C_1=\sum_{i=1}^{m/2}\Big[\kappa_{ix,ix}(w_{1,i}\cos^2(\mu_{\theta i})+w_{2,i}) \\ +\kappa_{iy,iy}(w_{1,i}\sin^2(\mu_{\theta i})+w_{2,i})+2\,\kappa_{ix,iy}(w_{1,i}\cos(\mu_{\theta i})\sin(\mu_{\theta i}))\Big] \tag{7.16}$$

where the values of $w_{k,i}$ are defined as:

$$w_{1,i}=f_i^2\left(\exp(-2\,\sigma_{\theta i}^2)-\exp(-\sigma_{\theta i}^2)\right) \tag{7.17}$$

$$w_{2,i}=f_i^2\left(1-\exp(-2\,\sigma_{\theta i}^2)\right)/2 \tag{7.18}$$

Next, a set of n load cases are defined by applying a load of unit magnitude at the deterministic location and in the mean direction of each uncertain load in turn. The compliance value computed for each of these load cases, using (7.11), is multiplied by $w_{1,i}$ producing:

$$C_{2,i}(1,\mu_{\theta i})=w_{1,i}\Big[\kappa_{ix,ix}(\cos^2(\mu_{\theta i}))+\kappa_{iy,iy}(\sin^2(\mu_{\theta i})) \\ +2\,\kappa_{ix,iy}(\cos(\mu_{\theta i})\sin(\mu_{\theta i}))\Big] \tag{7.19}$$

Subtracting the sum of compliance values calculated by (7.19) for all loads with uncertain applied direction from (7.16) results in:

$$E[C]-C_1-\sum_{i=1}^{n}C_{2,i}=\sum_{i=1}^{m/2}w_{2,i}\left[\kappa_{ix,ix}+\kappa_{iy,iy}\right] \tag{7.20}$$

Finally, two further sets of load cases are required to complete the computation of expected compliance. These load cases are defined by applying a load of unit magnitude at the location of each uncertain load in the horizontal ($\theta_x = 0$ or π) and vertical ($\theta_y = \pi/2$ or $3\pi/2$) directions separately. The compliance values for these two load cases are computed for each uncertain load using (7.11) and multiplied by $w_{2,i}$ producing:

$$C_{x,i}(1,\theta_x) = w_{2,i}\kappa_{ix,ix}$$
$$C_{y,i}(1,\theta_y) = w_{2,i}\kappa_{iy,iy} \tag{7.21}$$

This completes the computation of expected compliance considering uncertainty in loading direction (7.12) using a series of separate load cases. In the general case a maximum of 1+3*n* load cases is required where *n* is the number of loads with uncertain direction. The first load case is the application of deterministic loads multiplied by exponential functions of the variance (7.13) and each subsequent set of three load cases are applied at the location of the uncertain load. The subsequent loads have a magnitude of one and are applied in the mean, horizontal and vertical directions in turn. Using (7.15, 19, 21) the expected compliance becomes a weighted sum of the 1+3*n* load cases:

$$\begin{aligned} E[C(f,\theta)] = C_1(\bar{f},\mu_\theta) + \sum_{i=1}^{n}\Big[& w_{1,i}C_{2,i}(1,\mu_{\theta i}) \\ & + w_{2,i}\left(C_{x,i}(1,\theta_x) + C_{y,i}(1,\theta_y)\right)\Big] \end{aligned} \tag{7.22}$$

The number of load cases can be reduced by one for every value of $\mu_{\theta i}$ that is equal to θ_x or θ_y by simply combining load cases and weights appropriately. Also if all values of $\sigma_{\theta,i}$ are zero, then (7.22) reduces to a single load case equal to the compliance of the deterministic problem (1.1).

7.4 Combined loading uncertainty

A combined uncertainty formulation for expected compliance is constructed by introducing magnitude uncertainty into the formulation for expected compliance under directional uncertainty (7.22) using the result of (7.7). Starting with (7.22) the inclusion of loading magnitude uncertainties yields:

$$E[C(f,\theta)] = \int_{f_n} \cdots \int_{f_1} E[C(f,\theta)] \prod_{i=1}^{n} P(f_i)\, df_1 \cdots df_n \tag{7.23}$$

The results of (7.7) and (7.22) are used to evaluate (7.23) for each uncertain magnitude between limits $\mu_i \pm \zeta$. Letting ζ go to infinity results in an equation with the same format as (7.22), with the following modified weights that account for the uncertainty in loading magnitude:

$$\bar{f}_i = \mu_i \exp(-\sigma_{\theta i}^2/2) \tag{7.24}$$

$$w_{1,i} = \mu_i^2 \left(\exp(-2\sigma_{\theta i}^2) - \exp(-\sigma_{\theta i}^2) \right) + \sigma_i^2 \exp(-2\sigma_{\theta i}^2) \tag{7.25}$$

$$w_{2,i} = (\mu_i^2 + \sigma_i^2) \left(1 - \exp(-2\sigma_{\theta i}^2) \right) / 2 \tag{7.26}$$

Including uncertainties in magnitude does not increase the number of load cases required to compute expected compliance when directional uncertainties are already included. Also, if all values of σ_i are zero then the expected compliance in the presence of just directional uncertainties is recovered (7.22). Correspondingly, if all values of $\sigma_{\theta i}$ are zero, then the expected compliance in the presence of exclusively loading magnitude uncertainties is also recovered (7.7). Therefore, (7.22) with weights defined by (7.24-26) computes an accurate value for the expected compliance using a maximum of 1+3*n* load cases, where *n* loads can have uncertainties in both magnitude and direction. This expression for expected compliance is used to transform the robust optimisation problem (7.1) into an equivalent multiple load case one. This formulation can be implemented more efficiently by exploiting the linear elastic nature of the structure to reduce the number of load vectors needed when computing the displacement vectors for the required load cases (Conti et al. 2008). In the general case 2*n* load vectors could be used to compute displacements for the 1+3*n* load cases.

The minimisation of compliance problem with multiple loads cases can be solved by most existing methods (Bendsøe & Sigmund 2004; Allaire & Jouve

2005). Thus, the formulation for expected compliance obtained here is also valid for any computational topology optimisation method that can solve the multiple load case problem. When using the level set based method, the shape sensitivity for the multiple load case problem, ς_m is simply the sum of sensitivities from each load case (Allaire & Jouve 2005):

$$\varsigma_m = \sum_{i=1}^{n} w_i \tilde{E}\, \varepsilon(u_i)\, \varepsilon(u_i) \tag{7.27}$$

where n is the number of load cases and w_i are weights. This sensitivity definition is directly used to construct the velocity function for the level set optimisation method in the same manner as a single load case problem (2.17):

$$V_n = \lambda - \varsigma_m \tag{7.28}$$

The analytical formulation derived in this work (7.22) only requires 1+3n load cases to accurately compute expected compliance and the required sensitivities. Therefore, the computational cost of computing the objective and sensitivities scales linearly with the number of loads with magnitude and directional uncertainty. This is seen as more efficient than existing methods that rely on discretization to define the required load cases. For example, if there are n loading uncertainties, including both magnitude and direction, and S is the number of discrete samples used per uncertain variable, a total of S^n load cases could be required to compute the objective and sensitivities.

7.5 Examples

The minimisation of expected compliance problem under loading uncertainty is solved for a range of examples and compared to deterministic solutions. Expected compliance is defined using multiple load cases (7.22) with weights defined by (7.24-26). Problems are solved using the direct level set method detailed in

Chapter 5 with the hole insertion method created in Chapter 6, where the velocity function for the multiple load case problem is defined by (7.28). The hole insertion parameters used for all examples are: $\omega = 6h$, $\overline{h} = h$, $\overline{Vol} = 2\%$. Young's Modulus and Poisson's ratio of 1.0 and 0.3 are used for all examples and design domains are discretized using unit sized square elements.

7.5.1 Simple column

The simple column structure shown in Figure 7-1a is optimized for deterministic and uncertain loading conditions. The single point load f has mean magnitude $\mu = 10$ and mean applied direction $\mu_\theta = 3\pi/2$. To observe the effect of directional uncertainty on this simple example, the minimum expected compliance problem is solved for a range of applied direction standard deviations, $\sigma_\theta = 0.0$, 0.1, 0.2, 0.3, where $\sigma_\theta = 0.0$ is the deterministic case. The volume constraint for each problem is set to 20% of the design domain and a termination criterion of $\gamma=6\times10^{-4}$ is employed. For this example the inclusion of directional uncertainty only requires two load cases to accurately compute the expected compliance each iteration. The first load case is a unit load applied vertically with a weight of $50\times[1+\exp(-2\sigma_\theta^2)]$ and the second load case is a unit load applied horizontally with a weight of $50\times[1-\exp(-2\sigma_\theta^2)]$. These weights are calculated using (7.24-26)

The deterministic solution is a straight column shown in Figure 7-1b where all the material is aligned to support the vertically applied load. The expected compliance values for the deterministic solution under the uncertain loading conditions are 1048, 2242, 4079 for $\sigma_\theta = 0.1$, 0.2, 0.3, respectively. This result demonstrates that expected compliance for a deterministic solution can increase as loading directional uncertainty increases.

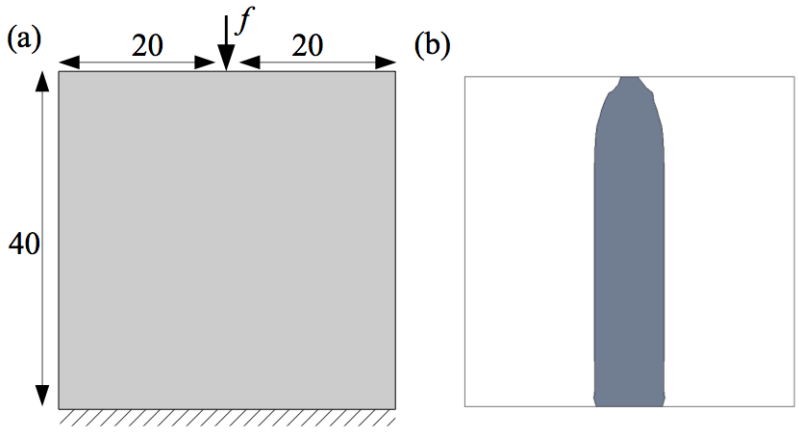

Figure 7-1: Simple column. a) Design domain and boundary conditions, b) Deterministic solution.

The solutions from the minimisation of expected compliance problem produce structures where the column splits to form arches, Figure 7-2a-c. These designs are more robust against loading directional uncertainty. The values for expected compliance are 752 for σ_θ = 0.1, 900 for σ_θ = 0.2 and 1050 for σ_θ = 0.3, demonstrating improved performance over the deterministic solution. It also appears that the solutions from the robust optimisation problem adapt to the level of uncertainty, as the width of the arch structure increases with increased variance. This simple example shows that more robust solutions can be obtained in the presence of loading direction uncertainty, as the expected compliance is significantly lower for the robust solutions compared with the deterministic one.

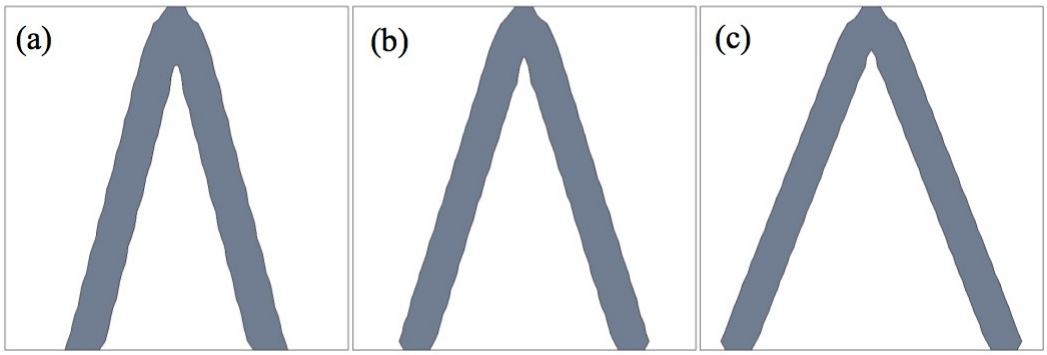

Figure 7-2: Simple column robust solutions. a) σ_θ = 0.1, b) σ_θ = 0.2, c) σ_θ = 0.3.

7.5.2 Carrier plate

A simple carrier plate example, Figure 7-3a is used to demonstrate that the approach developed in this work for including uncertainty in loading can also be applied to distributed loads. The uniformly distributed load F has mean magnitude $\mu = 1.0$ / unit length, mean applied direction $\mu_\theta = 3\pi/2$ and standard deviation $\sigma_\theta = 0.25$. The design domain is discretized using 100×100 elements and the bottom and top two rows of elements are fixed to remain part of the structure. This ensures the structure remains attached to the loading and boundary conditions during optimisation. The problem is solved for deterministic ($\sigma_\theta = 0.0$) and uncertain loading conditions. For this example the robust problem only requires two load cases. The first load case is a vertically applied uniformly distributed load with a magnitude of 1.0 / unit length with a weighting of 0.941. The second uniformly distributed load is applied in the horizontal direction along the top edge with a magnitude of 1.0 / unit length and weighting of 0.059, where the weights are calculated using (7.24-26). The volume constraint for both problems is set to 50% of the design domain and a termination criterion of $\gamma=10^{-3}$ is employed.

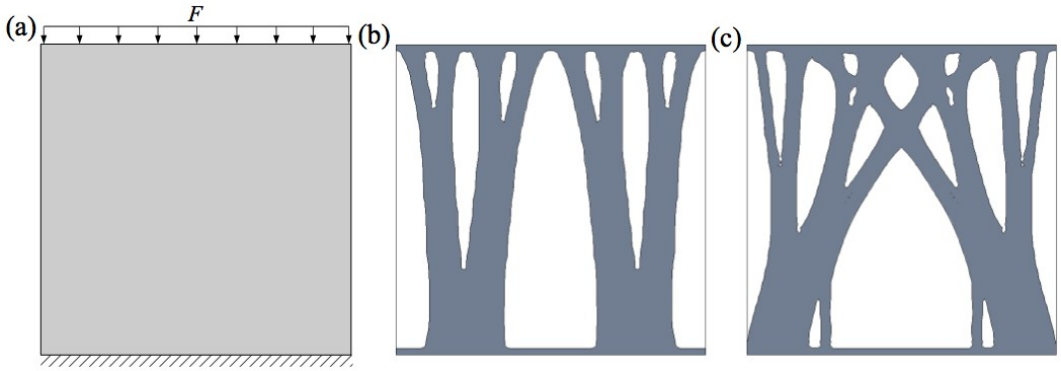

Figure 7-3: Carrier plate example. a) Design domain and boundary conditions, b) Deterministic solution ($\sigma_\theta = 0.0$), c) Robust solution ($\sigma_\theta = 0.25$).

The deterministic solution, Figure 7-3b, contains members that are mainly aligned to withstand the vertical uniformly distributed load. The deterministic compliance for this solution is 21.4×10^3, however, when the directional

uncertainty is considered for the deterministic solution, the expected compliance is significantly larger, 77.7×10^3. The robust solution depicted in Figure 7-3c still contains some vertical members, but also possesses more diagonal members able to cope with the uncertainty in the applied direction of the loading. The expected compliance for the robust solution is reduced to 33.6×10^3 demonstrating a significant increase in performance under uncertain loading direction compared with the deterministic solution.

7.5.3 Mast structure

A mast structure, Figure 7-4a, is designed for deterministic and uncertain loading conditions. The two vertical loads are considered without uncertainty and both have a magnitude f_1 = 5.0. For this example the uncertainty exists in the magnitude of the horizontal side loads, where the uncertainty is modelled by a half-normal probability distribution:

$$P(x)=\left(2/\pi\rho^2\right)^{1/2}\exp\left(-x^2/2\rho^2\right)\ ,\quad x\in[0,\infty) \tag{7.29}$$

where ρ is the shape parameter of the distribution. Substituting (7.29) into (7.4) and evaluating between limits 0 to ∞ using the same integration by parts approach detailed in Section 7.2 yields the following equation for expected compliance:

$$E[C]=\sum_{i=1}^{n}\kappa_{i,i}\rho_i^2 \tag{7.30}$$

Therefore, an additional load case of magnitude ρ_i is required for each magnitude uncertainty described by a half-normal distribution. This approach can be used to model uncertain loads that follow a Gaussian distribution, but are restricted to one sign, for example pressure loads such as wind and snow loading and reaction loads arising from tension cables. Two uncertain loading conditions are considered for the uniformly distributed side loads, f_2 and f_3, shown in Figure

7-4a. The first condition assumes both loads are described by identical probability distributions, $\rho_2 = \rho_3 = 0.5$. The second condition assumes the left side load has a lower mean, where $\rho_2 = 0.1$ and $\rho_3 = 0.5$. The volume constraint for each problem is set to 30% of the design domain and the termination criterion is $\gamma = 5\times10^{-4}$.

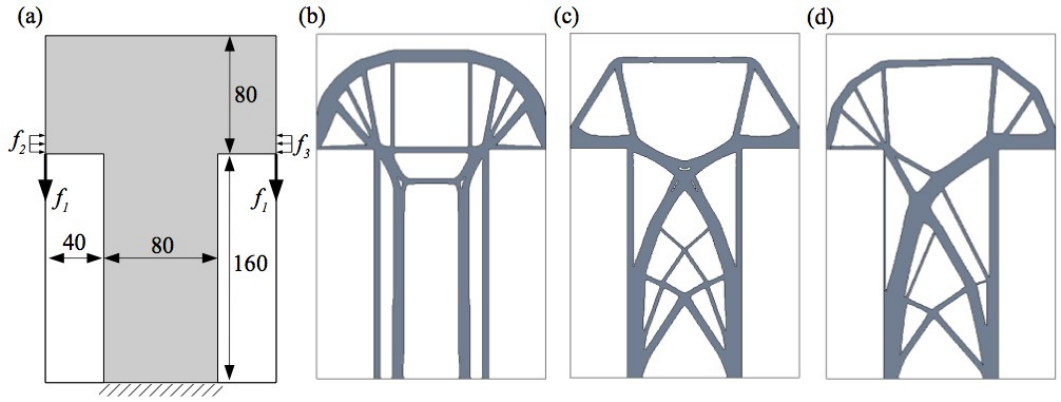

Figure 7-4: Mast example. a) Design domain boundary conditions, b) Deterministic solution, c) Robust solution ($\rho_2 = \rho_3 = 0.5$), d) Robust solution ($\rho_2 = 0.1$, $\rho_3 = 0.5$).

The deterministic solution, without considering the side loads, Figure 7-4b, contains vertical members in the central column aligned to carry the vertical loads. The deterministic compliance for this solution is 1.72×10^3, whereas the expected compliance considering the uncertain loads is 241.7×10^3 and 126.5×10^3 for the first and second uncertain conditions, respectively.

The robust solution for the first uncertain loading condition, Figure 7-4c, is achieved using just three load cases during each iteration. The solution is similar to the deterministic one, except that the central column has been reinforced with a cross braced structure and the top section contains fewer structural components. This structural configuration helps to significantly lower the expected compliance, when compared with the deterministic solution, to 9.43×10^3.

The second uncertain loading condition is now considered and the robust solution shown in Figure 7-4d is obtained. This solution is similar to the first robust solution. However, there is an increased reinforcement to support the greater uncertain loading applied on the right hand side. This distribution of

material produces an expected compliance of 5.75×10^3, which is again a significant improvement compared with the deterministic solution. This example demonstrates that a significant improvement in structural solutions with a marked difference in topology can be obtained when uncertain load cases are considered during optimisation.

7.5.4 Combined uncertainty example

A double hook structure, Figure 7-5a, is designed under deterministic and uncertain loading conditions where both loads have magnitude and directional uncertainties. For both loads f_1 and f_2, probability data are as follows: mean magnitude μ = 5.0 and standard deviation σ = 0.5, mean loading direction μ_θ = $3\pi/2$ and standard deviation σ_θ= 0.25. Five load cases as defined by (7.22) are thus required to compute expected compliance with weights calculated using (7.24-26). The first of these load cases is the simultaneous application of two vertical loads at the two loading points, where both loads have a magnitude of 4.846 and the load case weight is 1.0. The second case is a unit horizontal load applied at loading point one with a weight of 0.2797 and the third is a unit vertical load with weight of 1.483. The set of five load cases is completed by repeating the previous two load cases (with identical weights) for loading point two. The volume constraint for both problems is set to 50% of the design domain and the termination criterion is $\gamma=10^{-3}$.

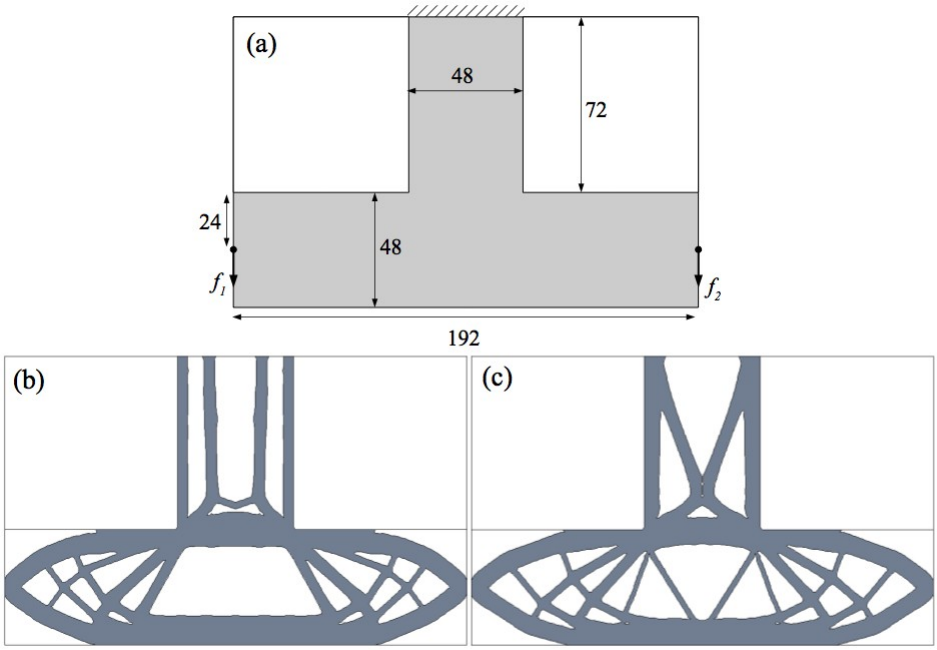

Figure 7-5: Double hook example. a) Design domain and boundary conditions, b) Deterministic solution, c) Robust solution.

The deterministic problem produces a solution with deterministic compliance of 26.23×10^2, Figure 7-5b. However, when uncertain loading conditions are applied to this solution, the expected compliance increases to 49.12×10^2. The robust solution is similar to the deterministic one except for the design of the central beam section, which is reinforced by cross braces, Figure 7-5c. It is clear that this beam in the deterministic solution essentially acts like a tie-bar predominantly in tension, but the robust solution recognises the bending due to the uncertain loading conditions. The expected compliance of the robust solution is 29.43×10^2, again demonstrating an improved performance compared with the deterministic problem in the presence of uncertain loading conditions.

It is interesting to note that if only magnitude uncertainties are considered, then expected compliance for the deterministic solution is 29.0×10^2. However, if only directional uncertainty is considered, expected compliance is 46.3×10^2. This suggests that, for this example, the compliance of the deterministic solution is more sensitive to directional uncertainty compared to magnitude uncertainty.

7.6 Conclusions

Expected compliance was considered as a robust objective function when introducing uncertainty into the deterministic problem. Loading uncertainty was considered in both direction and magnitude, where uncertain variables were described by normal probability distributions. Analytical expressions were derived to transform the expected compliance into a total compliance of a multiple load case problem. For a general problem, only three additional cases per uncertain load were required to accurately compute compliance and sensitivities. Therefore, the approach was more efficient than existing discretisation methods.

The robust optimisation problem was solved for continuum structures using the level set method. The robust solutions show marked differences when compared with the equivalent deterministic cases. Furthermore, the expected compliance was significantly reduced compared with deterministic solutions. This demonstrated the importance of including uncertainties in structural topology optimisation methods to produce robust solutions.

Chapter 8

Including compliance variance under loading uncertainty

8.1 Introduction

Robustness in engineering design and optimisation has been described in a number of ways, but is generally considered to be the insensitivity of the design to variations (G-J. Park et al. 2006). Therefore, including compliance variance in structural topology optimisation is important to produce robust designs that are reasonably insensitive to the uncertain operating environment. The variance of a function, $Var[g(x)]$ can be calculated by:

$$Var[g(x)] = E[g(x)^2] - E[g(x)]^2 \tag{8.1}$$

where $g(x)$ is a function with uncertain variables x.

The desirability of extending the minimisation of expected compliance problem under loading uncertainty to include variance has been highlighted by some authors (Alvarez & Carrasco 2005; Guest & Igusa 2008). However, to the author's knowledge, no method has yet been fully developed and implemented.

This chapter extends the analytical approach developed in the previous chapter to solve the minimisation of expected compliance problem to include compliance variance. The method developed here considers only uncertainty in loading magnitude.

8.2 Loading magnitude uncertainty

This section derives an analytical expression, avoiding simplification or approximation, for the variance of the compliance objective function (1.1) when there is uncertainty in the magnitude of the applied loads. Again it is assumed uncertain loading magnitudes are described by normal distributions. The variance of compliance can be defined using (8.1):

$$Var[C(f)] = E[C(f)^2] - E[C(f)]^2 \tag{8.2}$$

where the compliance function, $C(f)$ is written in the following discrete form:

$$C(f) = \sum_{i,\,j=1}^{m/2} f_i f_j \left(\kappa_{ix,\,jx} \cos(\theta_i)\cos(\theta_j) + \kappa_{iy,\,jy} \sin(\theta_i)\sin(\theta_j) + 2\kappa_{ix,\,jy} \cos(\theta_i)\sin(\theta_j) \right) \tag{8.3}$$

where $\kappa_{i,j}$ are entries in the inverse stiffness matrix, m is the matrix order, f_i are the uncertain loading magnitudes and θ_i are loading directions, which are considered certain and fixed in the following derivation.

8.2.1 Derivation

Compliance variance (8.2) is evaluated analytically by finding expressions for each part on the right hand side. First an expression for $E[C(f)^2]$ is found by evaluating:

$$E[C(f)^2] = \int_{f_n} \cdots \int_{f_1} C(f)^2 \prod_{i=1}^{n} P(f_i) df_1 \cdots df_n \tag{8.4}$$

where $P(f_i)$ are normal distributions of the uncertain loading magnitudes with mean, μ_i and standard deviation, σ_i. Using (8.3) as the compliance function, (8.4) is evaluated using integration by parts between limits, $\mu_i \pm \zeta$ for each load, then letting ζ go to infinity yields:

$$\begin{aligned} E[C(f)^2] = & \sum_{i,j,p,q=1}^{m/2} \mu_i \mu_j \mu_p \mu_q \Theta_1 + \sum_{i,j,p=1}^{m/2} \mu_i \mu_j \sigma_p^2 (4\Theta_2 + 2\Theta_3) \\ & + \sum_{i,j=1}^{m/2} \sigma_i^2 \sigma_j^2 (2\Theta_4 + \Theta_5) \end{aligned} \tag{8.5}$$

where the Θ_k are functions of inverse stiffness matrix entries, $\kappa_{i,j}$ and loading directions, θ_i. Further steps between (8.4) and (8.5) are detailed in Appendix 1.3.

$$\begin{aligned} \Theta_1 = & \kappa_{ix,jx}\kappa_{px,qx}\cos(\theta_i)\cos(\theta_j)\cos(\theta_p)\cos(\theta_q) \\ & + \kappa_{iy,jy}\kappa_{py,qy}\sin(\theta_i)\sin(\theta_j)\sin(\theta_p)\sin(\theta_q) \\ & + 4\kappa_{ix,jy}\kappa_{px,qy}\cos(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_q) \\ & + 2\kappa_{ix,jx}\kappa_{py,qy}\cos(\theta_i)\cos(\theta_j)\sin(\theta_p)\sin(\theta_q) \\ & + 4\kappa_{ix,jx}\kappa_{px,qy}\cos(\theta_i)\cos(\theta_j)\cos(\theta_p)\sin(\theta_q) \\ & + 4\kappa_{iy,xy}\kappa_{px,qy}\sin(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_q) \end{aligned} \tag{8.6}$$

$$
\begin{aligned}
\Theta_2 = {} & \kappa_{ix,px}\kappa_{jx,px}\cos(\theta_i)\cos(\theta_j)\cos^2(\theta_p) \\
& + \kappa_{iy,py}\kappa_{jy,py}\sin(\theta_i)\sin(\theta_j)\sin^2(\theta_p) \\
& + 2\kappa_{ix,py}\kappa_{px,jy}\cos(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_p) \\
& + \kappa_{px,iy}\kappa_{px,jy}\sin(\theta_i)\sin(\theta_j)\cos^2(\theta_p) \\
& + \kappa_{ix,py}\kappa_{jx,py}\cos(\theta_i)\cos(\theta_j)\sin^2(\theta_p) \\
& + 2\kappa_{ix,px}\kappa_{jy,py}\cos(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_p) \\
& + 2\kappa_{ix,px}\kappa_{px,jy}\cos(\theta_i)\sin(\theta_j)\cos^2(\theta_p) \\
& + 2\kappa_{ix,px}\kappa_{jx,py}\cos(\theta_i)\cos(\theta_j)\cos(\theta_p)\sin(\theta_p) \\
& + 2\kappa_{iy,py}\kappa_{jx,py}\sin(\theta_i)\cos(\theta_j)\sin^2(\theta_p) \\
& + 2\kappa_{iy,py}\kappa_{px,jy}\sin(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_p)
\end{aligned}
\tag{8.7}
$$

$$
\begin{aligned}
\Theta_3 = {} & \kappa_{ix,jx}\kappa_{px,px}\cos(\theta_i)\cos(\theta_j)\cos^2(\theta_p) \\
& + \kappa_{iy,jy}\kappa_{py,py}\sin(\theta_i)\sin(\theta_j)\sin^2(\theta_p) \\
& + 4\kappa_{ix,jy}\kappa_{px,py}\cos(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_p) \\
& + \kappa_{px,px}\kappa_{iy,jy}\sin(\theta_i)\sin(\theta_j)\cos^2(\theta_p) \\
& + \kappa_{ix,jx}\kappa_{py,py}\cos(\theta_i)\cos(\theta_j)\sin^2(\theta_p) \\
& + 2\kappa_{px,px}\kappa_{ix,jy}\cos(\theta_i)\sin(\theta_j)\cos^2(\theta_p) \\
& + 2\kappa_{ix,jx}\kappa_{px,py}\cos(\theta_i)\cos(\theta_j)\cos(\theta_p)\sin(\theta_p) \\
& + 2\kappa_{py,py}\kappa_{ix,jy}\cos(\theta_i)\sin(\theta_j)\sin^2(\theta_p) \\
& + 2\kappa_{iy,jy}\kappa_{px,py}\sin(\theta_i)\sin(\theta_j)\cos(\theta_p)\sin(\theta_p)
\end{aligned}
\tag{8.8}
$$

$$
\begin{aligned}
\Theta_4 = {} & \kappa_{ix,jx}^2\cos^2(\theta_i)\cos^2(\theta_j) + \kappa_{iy,jy}^2\sin^2(\theta_i)\sin^2(\theta_j) \\
& + 2\kappa_{ix,jy}^2\cos^2(\theta_i)\sin^2(\theta_j) \\
& + 2\kappa_{ix,jy}\kappa_{jx,iy}\cos(\theta_i)\sin(\theta_i)\cos(\theta_j)\sin(\theta_j) \\
& + 2\kappa_{ix,jx}\kappa_{iy,jy}\cos(\theta_i)\sin(\theta_i)\cos(\theta_j)\sin(\theta_j) \\
& + 4\kappa_{ix,jx}\kappa_{ix,jy}\cos^2(\theta_i)\cos(\theta_j)\sin(\theta_j) \\
& + 4\kappa_{iy,jy}\kappa_{ix,jy}\sin^2(\theta_j)\cos(\theta_i)\sin(\theta_i)
\end{aligned}
\tag{8.9}
$$

$$
\begin{aligned}
\Theta_5 = {} & \kappa_{ix,ix}\kappa_{jx,jx}\cos^2(\theta_i)\cos^2(\theta_j) \\
& + \kappa_{iy,iy}\kappa_{jy,jy}\sin^2(\theta_i)\sin^2(\theta_j) \\
& + 4\kappa_{ix,iy}\kappa_{jx,jy}\cos(\theta_i)\sin(\theta_i)\cos(\theta_j)\sin(\theta_j) \\
& + 2\kappa_{ix,ix}\kappa_{jy,jy}\cos^2(\theta_i)\sin^2(\theta_j) \\
& + 4\kappa_{ix,ix}\kappa_{jx,jy}\cos^2(\theta_i)\cos(\theta_j)\sin(\theta_j) \\
& + 4\kappa_{iy,iy}\kappa_{jx,jy}\sin^2(\theta_i)\cos(\theta_j)\sin(\theta_j)
\end{aligned}
\tag{8.10}
$$

Expected compliance under loading magnitude uncertainty is derived in Section 7.2. For loads applied in arbitrary directions (7.7) can be written as:

$$E[C(f)] = \sum_{i,j=1}^{m/2} \mu_i \mu_j \left(\kappa_{ix,jx} \cos(\theta_i)\cos(\theta_j) + \kappa_{iy,jy} \sin(\theta_i)\sin(\theta_j) + 2\kappa_{ix,jy} \cos(\theta_i)\sin(\theta_j) \right) + \sum_{i=1}^{m/2} \sigma_i^2 \left(\kappa_{ix,ix} \cos^2(\theta_i) + \kappa_{iy,jy} \sin^2(\theta_i) + 2\kappa_{ix,iy} \cos(\theta_i)\sin(\theta_i) \right) \tag{8.11}$$

The second part of the compliance variance function (8.2) is simply evaluated by squaring (8.11):

$$E[C(f)]^2 = \sum_{i,j,p,q=1}^{m/2} \mu_i \mu_j \mu_p \mu_q \Theta_1 + 2 \sum_{i,j,p=1}^{m/2} \mu_i \mu_j \sigma_p^2 \Theta_3 + \sum_{i,j=1}^{m/2} \sigma_i^2 \sigma_j^2 \Theta_5 \tag{8.12}$$

Now an exact analytical expression for compliance variance under loading magnitude uncertainty can be derived from (8.2, 5, 12):

$$Var[C(f)] = 4 \sum_{i,j,p=1}^{m/2} \mu_i \mu_j \sigma_p^2 \Theta_2 + 2 \sum_{i,j=1}^{m/2} \sigma_i^2 \sigma_j^2 \Theta_4 \tag{8.13}$$

This expression can be re-written in matrix vector form:

$$Var[C(f)] = 4\{\mu\}^T [K]^{-1} [\bar{\sigma}] [K]^{-1} \{\mu\} + 2 \sum_{i=1}^{n} \{\sigma_i\}^T [K]^{-1} [\bar{\sigma}] [K]^{-1} \{\sigma_i\} \tag{8.14}$$

where $[K]$ is a square symmetric stiffness matrix, $\{\mu\}$ is the mean or deterministic loading vector and $\{\sigma_i\}$ is a load vector with a single load whose magnitude is equal to the standard deviation of uncertain load *i*. The matrix $[\bar{\sigma}]$ acts like a covariance matrix and is defined as:

$$[\bar{\sigma}]=\begin{bmatrix} \ddots & 0 & 0 & 0 \\ 0 & \sigma_i^2\cos^2(\theta_i) & \sigma_i^2\cos(\theta_i)\sin(\theta_i) & 0 \\ 0 & \sigma_i^2\cos(\theta_i)\sin(\theta_i) & \sigma_i^2\sin^2(\theta_i) & 0 \\ 0 & 0 & 0 & \ddots \end{bmatrix} \tag{8.15}$$

Conveniently the load vectors required to compute compliance variance using (8.14) are identical to those required to compute expected compliance using (8.11).

8.2.2 Sensitivity analysis

This section derives shape sensitivities for the compliance variance function (8.14). These sensitivities can then be used with the level set optimisation method detailed in Chapters 5 and 6 to solve robust topology optimisation problems involving variance of compliance. First (8.14) is re-written in terms of displacements using the finite element equation:

$$Var[C(f)]= 4\{u_\mu\}^T[\bar{\sigma}]\{u_\mu\}+2\sum_{i=1}^{n}\{u_i\}^T[\bar{\sigma}]\{u_i\} \tag{8.16}$$

where $\{u_\mu\}$ and $\{u_i\}$ are displacement vectors for the load vectors $\{\mu\}$ and $\{\sigma_i\}$ respectively:

$$\begin{aligned} \{\mu\}&=[K]\{u_\mu\} \\ \{\sigma_i\}&=[K]\{u_i\} \end{aligned} \tag{8.17}$$

The formulation for compliance variance (8.16) can be moved from a discrete setting to a continuous one:

$$Var[C(f)]= 4\int_{\Gamma_s}\bar{\sigma}\,u_\mu u_\mu\,d\Gamma_S+2\sum_{i=1}^{n}\int_{\Gamma_s}\bar{\sigma}\,u_i u_i\,d\Gamma_S \tag{8.18}$$

where $\bar{\sigma}$ is a tensor form of (8.15) that only has non-zero values where surface

traction loading is applied at the structure boundary, Γ_S. Thus, $\bar{\sigma}u$ acts like a surface traction loading. The displacement fields u_μ and u_i are solutions to the following equilibrium equations:

$$\int_{\Omega_S} \tilde{E}\,\varepsilon(u_\mu)\varepsilon(v)\,d\Omega_S = \int_{\Gamma_S} \mu v\,d\Gamma_S \tag{8.19a}$$

$$\int_{\Omega_S} \tilde{E}\,\varepsilon(u_i)\varepsilon(v)\,d\Omega_S = \int_{\Gamma_S} \sigma_i v\,d\Gamma_S \tag{8.19b}$$

where μ is the mean surface traction loading, σ_i is the standard deviation loading for uncertain load i and v is any kinematically admissible displacement field. Following the work of Allaire et al (2004) and noticing that $\bar{\sigma}$ is zero along the portion of the structure boundary free of surface tractions, Γ_0 the shape sensitivity can be determined using (2.10):

$$Var'[C(f)] = \int_{\Gamma_0} \left[4\tilde{E}\,\varepsilon(u_\mu)\varepsilon(p_\mu) + 2\sum_{i=1}^{n} \tilde{E}\,\varepsilon(u_i)\varepsilon(p_i) \right] V_n\,d\Gamma_0 \tag{8.20}$$

where p_μ and p_i are the solutions to the following adjoint problems:

$$\int_{\Omega_S} \tilde{E}\,\varepsilon(v)\varepsilon(p_\mu)\,d\Omega_S = \int_{\Gamma_S} 2\bar{\sigma}\,u_\mu v\,d\Gamma_S \tag{8.21a}$$

$$\int_{\Omega_S} \tilde{E}\,\varepsilon(v)\varepsilon(p_i)\,d\Omega_S = \int_{\Gamma_S} 2\bar{\sigma}\,u_i v\,d\Gamma_S \tag{8.21b}$$

Therefore, $2\bar{\sigma}u_\mu$ and $2\bar{\sigma}u_i$ are adjoint load vectors used to find the adjoint displacement vectors p_μ and p_i that are required to compute the shape sensitivity of compliance variance in (8.20). The velocity function in (8.20) is then simply defined to reduce the objective function in the same fashion as for the deterministic (2.17) and minimisation of expected compliance problems (7.28):

$$V_n = \lambda - \varsigma_v \tag{8.22}$$

where λ is a Lagrange multiplier for the volume constraint and:

$$\varsigma_v = 4\,\tilde{E}\,\varepsilon(u_\mu)\,\varepsilon(p_\mu) + 2\sum_{i=1}^{n}\tilde{E}\,\varepsilon(u_i)\,\varepsilon(p_i) \tag{8.23}$$

This velocity function can be used with the level set method developed in Chapters 5 and 6 to minimise the compliance variance of a structure under loading magnitude uncertainty, subject to a volume constraint.

8.3 Comparison to sampling technique

The formula derived for computing compliance variance (8.16) is compared to values computed using a sampling technique to check its validity and demonstrate its efficiency. The mean and variance of a function, $g(x)$ that has some uncertain variables, x that are described by continuous probability functions can be estimated using (G-J. Park et al. 2006):

$$\bar{E}[g(x)] = \sum_{i=1}^{n} g(x_i)\,P(x_i) \Big/ \sum_{i=1}^{n} P(x_i) \tag{8.24}$$

$$\overline{Var}[g(x)] = \sum_{i=1}^{n} P(x_i)\left|g(x_i) - \bar{E}[g(x)]\right|^2 \Big/ \sum_{i=1}^{n} P(x_i) \tag{8.25}$$

where n is the number of samples, $\bar{E}$ an estimate on expected value and $\overline{Var}$ an estimate of variance. Probability density functions are included in (8.24) and (8.25) as they were found to improve the estimation.

Approximate values for variance and expected compliance are compared with those computed using the analytically derived formulae (8.16) and (8.11) for a simple structure, Figure 8-1. The structure is discretised using 50 × 50 unit sized elements and analysed using the AFG method detailed in Chapter 3.

Load	μ_i	σ_i	θ_i (rad)
f_1	10.0	2.0	4.0
f_2	5.8	2.3	0.7
f_3	3.7	0.7	2.3

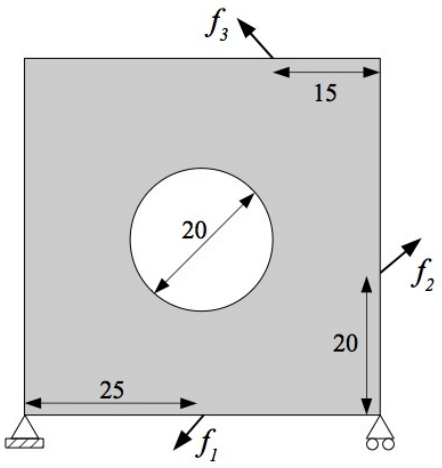

Figure 8-1: Simple test structure with uncertain loading conditions

First, only one load is considered, f_1 and the other two loads are initially ignored. Approximated values are computed using sample load cases that are $\mu_1 \pm m\sigma_1$, where m is an integer. The approximated values are improved by increasing m and including the new load cases generated in (8.24) and (8.25). The results for the single uncertain load show that when 5 sample load cases are used to approximate expected compliance the value converges to within 1% of the analytical value, which is computed using just two load cases, Table 8-1. However, it takes 7 sample load cases for the approximated variance to converge to within 1% of the analytical value, which is also computed using two load cases.

No. load cases	$E[C]$	$Var[C]$
1	722.7	0
3	738.5	46008.7
5	749.4	78385.7
7	751.5	84837.1
9	751.6	85222.6
Analytical	751.6	85230.1

Table 8-1: Test structure: approximated and analytical values for expected and variance of compliance, considering only load f_1.

Analytical and approximated values are now computed considering all three uncertain loads acting on the test structure, Figure 8-1. Sample load cases are again constructed using loads equal to $\mu_i \pm m\sigma_i$. However, to sample the interaction between loads, the number of sample load cases is equal to S^3, where S is the number of sample cases per uncertain load. The results show that for 125 sample load cases (5 per uncertain load) the approximated expected compliance is within 1% of the analytically computed value, which is computed using just 4 load cases, Table 8-2. However, for 125 sample load cases, there is an 11% difference between the approximated and analytical value for compliance variance. This suggests more sample load cases are required to accurately approximate compliance variance. However, adding another two samples per uncertain load increases the total number of sample load cases to 343.

No. load cases	$E[C]$	Var[C]
1	661.7	0
27	707.0	20487.5
125	738.2	37891.7
Analytical	744.5	42581.5

Table 8-2: Test structure: approximated and analytical values for expected and variance of compliance, considering all loads.

The results from the simple test structure show that the sampling method converges to the analytical values when a sufficient number of samples are used. However, the number of load case samples required to gain accurate computations is significantly greater than the number used to compute the analytical values, especially if there are many uncertain loads. Furthermore, more samples are required to accurately compute compliance variance compared to the expected value. This highlights the importance of the analytical approach to improve efficiency by minimising the number of load cases required to accurately compute statistical moments of compliance. This is particularly important in optimisation, where the objective function and sensitivities are computed each iteration.

8.4 Combined robust objective function

Robust structural optimisation problems often simultaneously consider both expected performance and variability of performance, (Doltsinis & Kang 2004; G-J. Park et al. 2006). This allows a structure to be designed with good mean performance and low variability about the mean. A common method for achieving such designs is to consider a weighted combination of expected and variance of performance, $Rb[g(x)]$ as the objective function:

$$Rb[g(x)] = \left(\frac{\eta}{w}\right) E[g(x)] + \left(\frac{1-\eta}{w^2}\right) Var[g(x)] \tag{8.26}$$

where $\eta \in [0, 1]$ is a weighting factor for the two parts of the objective and w is a non-dimensional weight with units the same as those of $g(x)$. The non-dimensional weight is necessary because variance of a function has units of the expected value squared. For the compliance objective, w can be defined using Young's modulus, E and the mean loading vector, μ:

$$w = (\mu^T \mu)/E \tag{8.27}$$

The disparity of units between expected and variance of a function can also be resolved using normalising factors. These factors are often the values computed from the initial design (G-J. Park et al. 2006). However, this approach can lead to solutions dependent on the initial design, thus the non-dimensional weight approach is adopted to avoid this problem. The ideal solution would be to replace the variance of the function with its standard deviation, thus avoiding the requirement of additional factors. However, computing sensitivities of the square root of compliance variance (8.16) is challenging.

The robust optimisation problem for compliance, considering both expected value and variance, is defined as:

$$\begin{aligned} &\text{Minimise: } Rb[C] = \left(\frac{\eta}{w}\right) E[C] + \left(\frac{1-\eta}{w^2}\right) Var[C] \\ &\text{Subject to: } \int_{\Omega} d\Omega \leq Vol^* \end{aligned} \tag{8.28}$$

where w is defined by (8.27). The shape sensitivity for the compliance combined robust objective function (8.28), $Rb'[C]$ is simply the combination of sensitivities for each part:

$$Rb'[C] = \left(\frac{\eta}{w}\right) E'[C] + \left(\frac{1-\eta}{w^2}\right) Var'[C] \tag{8.29}$$

where $E'[C]$ is defined as the sensitivity of the equivalent multiple load case problem defined in Chapter 7 and $Var'[C]$ is defined by (8.20). Thus, the velocity function for the combined objective is simply defined as:

$$V_n = \lambda - (\eta / w)\varsigma_m - \left((1-\eta)/w^2\right)\varsigma_v \tag{8.30}$$

where ς_m is defined by (7.27) for the equivalent multiple load formulation of expected compliance, ς_v is defined by (8.23) and λ is the Lagrange multiplier for the volume constraint.

8.5 Examples

The combined robust optimisation problem (8.28) is solved using the level set method detailed in Chapters 5 and 6 using the velocity function defined by (8.30). The effect of the combination weight, η is investigated, where η = 1 is the minimisation of expected compliance problem and η = 0 is the minimisation of compliance variance problem. Young's Modulus and Poisson's ratio of 1.0 and 0.3 are used for all examples.

8.5.1 Simple beam structure

The first example is a simple short beam structure with three point loads applied along the bottom edge, Figure 8-2a. The three loads all have the same mean magnitude, μ = 1.0, but have different standard deviations for magnitude uncertainty, σ_1 = 0.5, σ_2 = 0.1, σ_3 = 0.2. The design domain is discretised using 160 × 80 unit sized square elements and the volume constraint is 40% of the design domain. The convergence criterion is $\gamma = 5\times10^{-4}$. When the uncertain loading conditions are applied to the deterministic solution, Figure 8-2b, expected compliance equals 112.0 and variance is 1412.1.

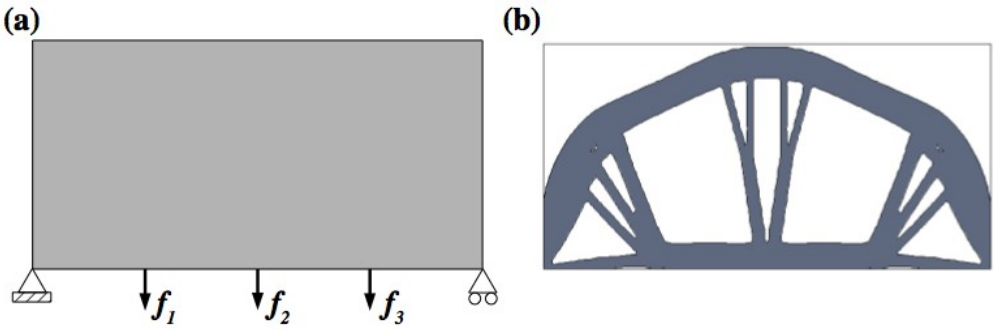

Figure 8-2: Simple beam example. a) Design domain and loading conditions, b) Deterministic solution.

Uncertainties are now considered during optimisation by solving the robust compliance problem (8.28) for various combination weights, η. Solutions for a range of combination weights are shown in Figure 8-3 and convergence histories are shown in Figure 8-4. All problems converge reasonably smoothly towards an optimum solution, which validates the sensitivity computation for the variance and combined objective functions. The smallest expected compliance value is for the solution using η = 1.0, Figure 8-3a, which also has the greatest compliance variance value. This is not surprising as the problem is effectively minimisation of expected compliance and variance is not considered during optimisation. Also, the minimum variance of compliance is for the variance minimisation problem, η = 0.0, Figure 8-3f. It is also interesting to note that, for this example, all solutions

that consider uncertainty during optimisation have significantly smaller compliance variance values compared to the deterministic solution. However, the same is not true of expected compliance values, where only the solution using η = 1.0 has a smaller value.

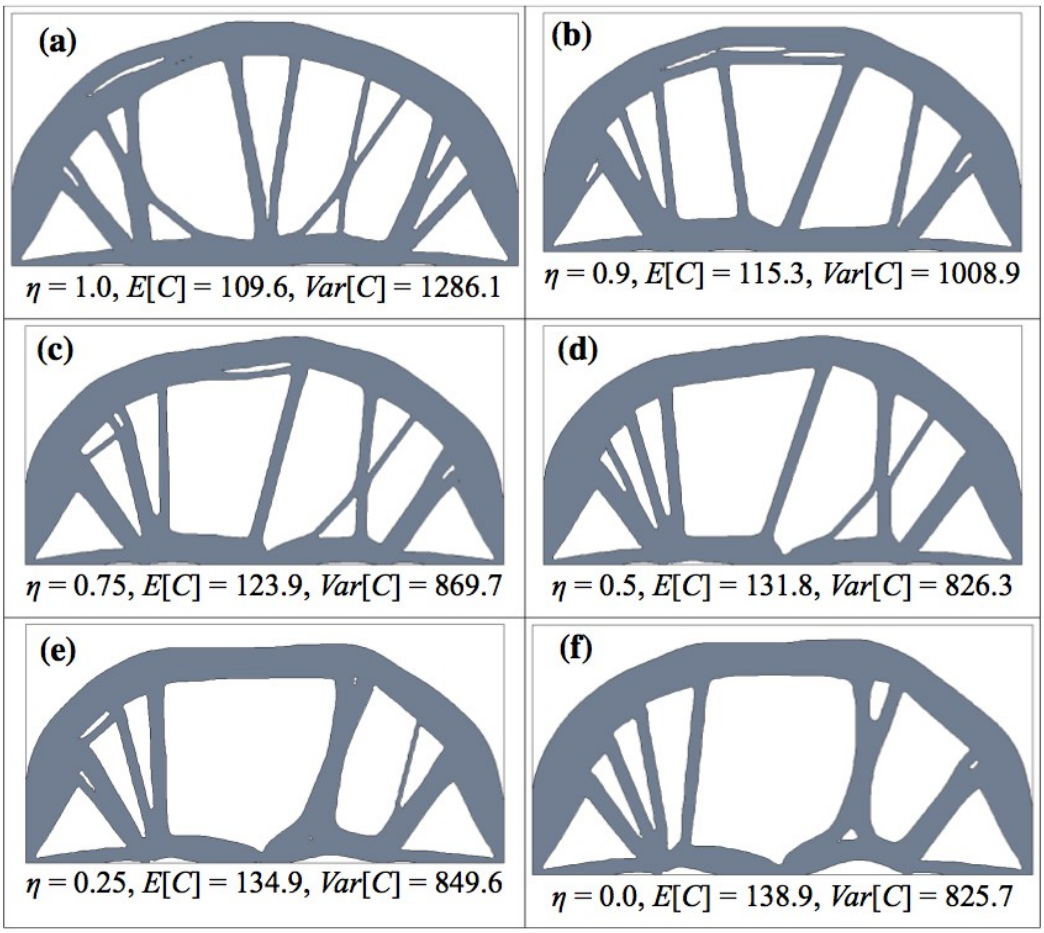

Figure 8-3: Robust solutions for the beam example using various combination weights.

For the beam example there appears to be a trade-off between expected compliance and variance of compliance. To investigate this further expected and variance values for a range of solutions are plotted against η, Figure 8-5. A clear trend is observed for $\eta \geq 0.5$, where the expected compliance decreases and variance increases as the combination weight increases. This seems reasonable, as the combined objective focuses more on expected compliance as the weight

increases (8.28). However, the trend for $\eta < 0.5$ is less clear, especially for the solution using $\eta = 0.4$, which has a larger expected compliance value than the solutions using combinations weights of 0.3 and 0.2.

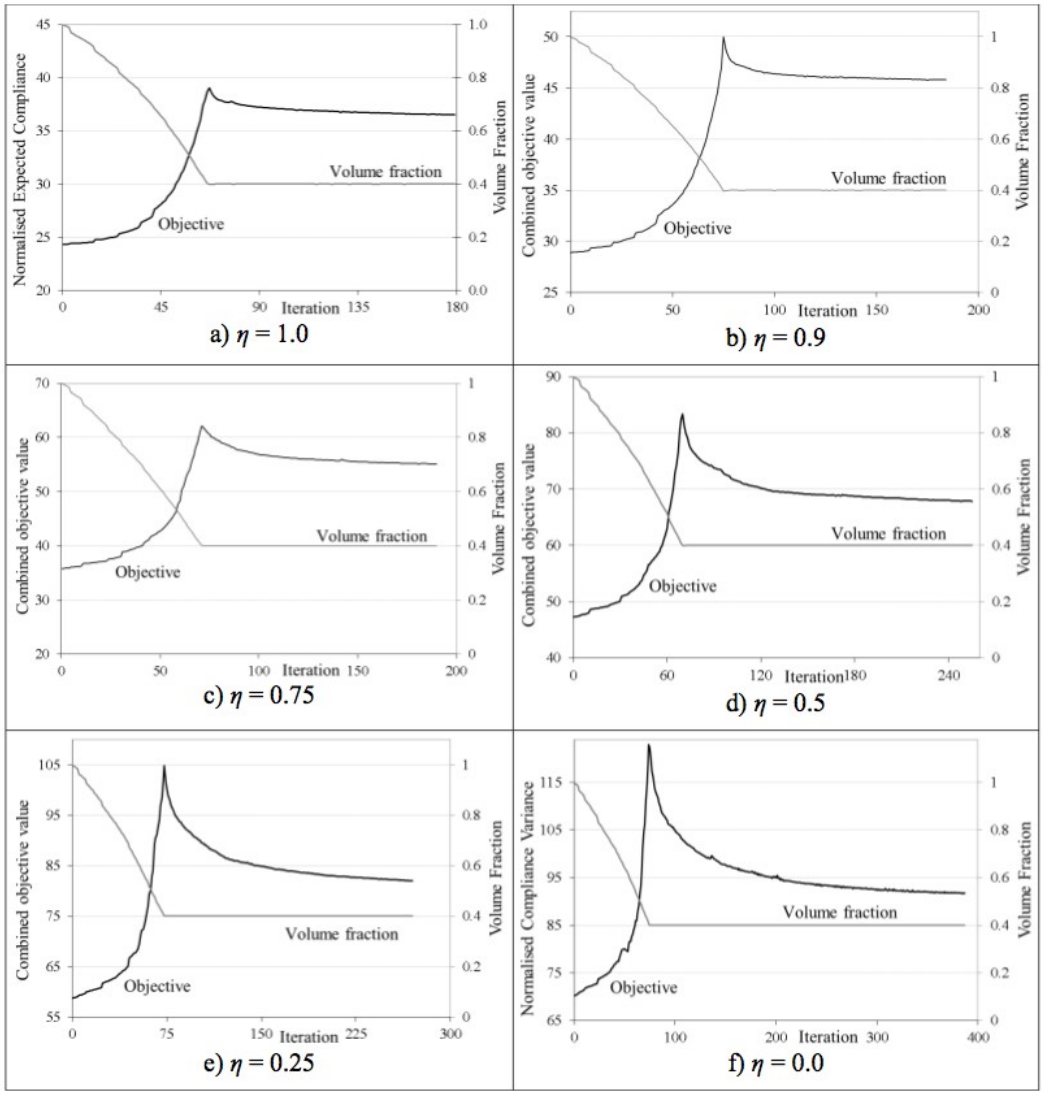

Figure 8-4: Convergence histories for the robust solutions of the beam example.

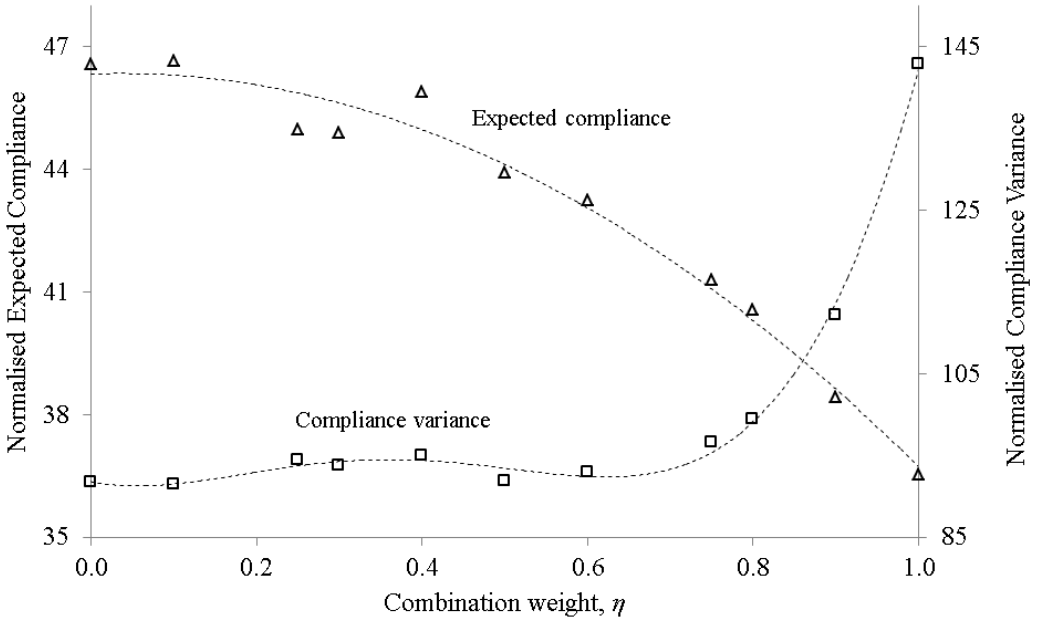

Figure 8-5: Beam example. Expected and variance of compliance for a range of combination weights.

The anomaly for $\eta = 0.4$ in Figure 8-5 appear to be a local minimum, where the central diagonal bar which is present in the solutions for $\eta \geq 0.5$ is eliminated, Figure 8-3 and Figure 8-6a. This is demonstrated by solving the same problem with a finer mesh of 320 × 160 unit sized elements. The solution using the finer mesh retains the central diagonal bar, Figure 8-6b.

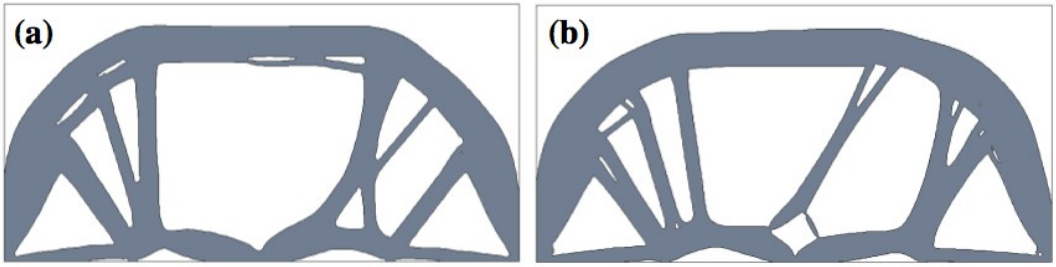

Figure 8-6: Beam example, solutions for $\eta = 0.4$. a) 160 × 80 mesh, b) 320 × 160 mesh.

8.5.2 Bridge structure

The second example is a bridge type structure, Figure 8-7a, where the uniformly distributed load applied along the top is split into two components with equal uncertainty in magnitude: μ = 0.1 / unit length and σ = 0.04 / unit length. The design domain is discretised using unit sized square elements and the volume constraint is 50% of the design domain. The convergence criterion is $\gamma = 1.1\times10^{-3}$. To ensure the loading does not become detached during optimisation, the top two rows of elements are fixed to remain part of the structure. When the uncertain loading conditions are applied to the deterministic solution, Figure 8-7b, expected compliance is 1320.6 and variance is 1326.0×10^{3}.

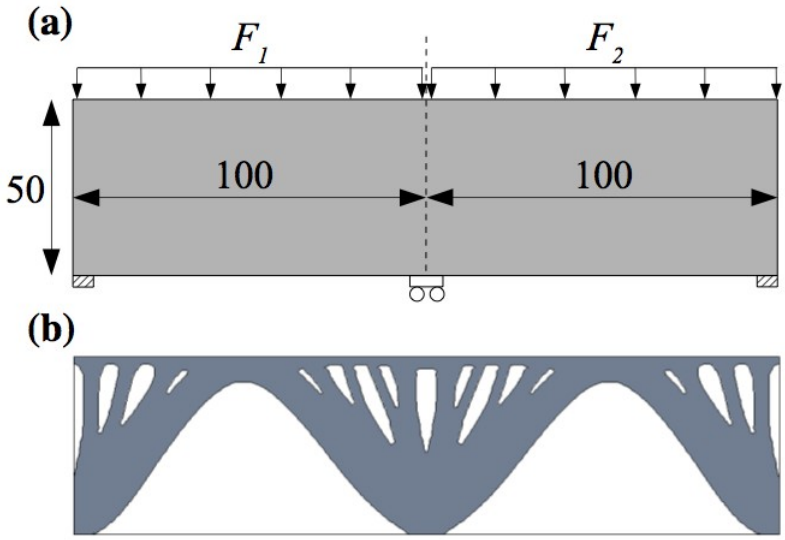

Figure 8-7: Bridge example. a) Design domain and loading conditions, b) Deterministic solution.

Uncertainty is now considered during optimisation and the solutions for η = 1.0, 0.5, 0.0 are shown in Figure 8-8a, b, c, respectively. Values for expected compliance are: 635.1, 633.7, 633.5 and values for compliance variance are: 100.7×10^{3}, 100.3×10^{3}, 100.2×10^{3}, for η = 1.0, 0.5, 0.0, respectively. All solutions

are similar to the deterministic solution, except for the addition of two lower horizontal bars. These bars arise to support the potentially unsymmetrical loading conditions and help significantly to reduce the expected value and variance of compliance, compared to the deterministic solution. Expected and variance values for a range of solutions are plotted against η, Figure 8-9.

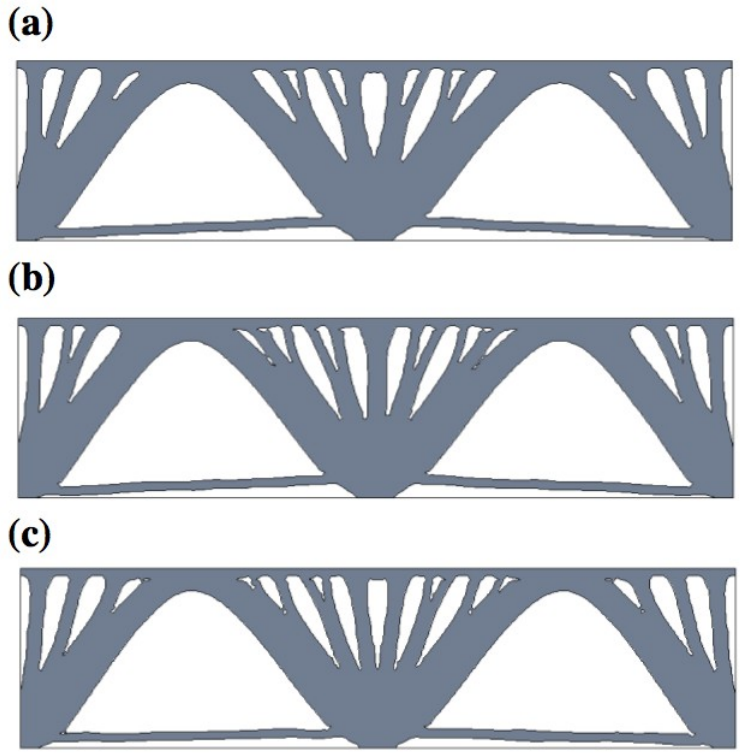

Figure 8-8: Bridge example solutions. a) Expected compliance objective, b) Combined objective η=0.5, c) Variance of compliance objective.

All solutions including uncertainty are very similar in design and have compliance values within 1%. This suggests that, for this problem, the expected and variance values of compliance are mutual, such that minimising one also minimises the other. This is in contrast to the previous example where there was a trade-off between expected and variance of compliance. The two examples demonstrate that the effect of introducing compliance variance into the objective is problem dependent. However, it is not clear what specific features of the two examples produce this contrasting behaviour when variance is introduced.

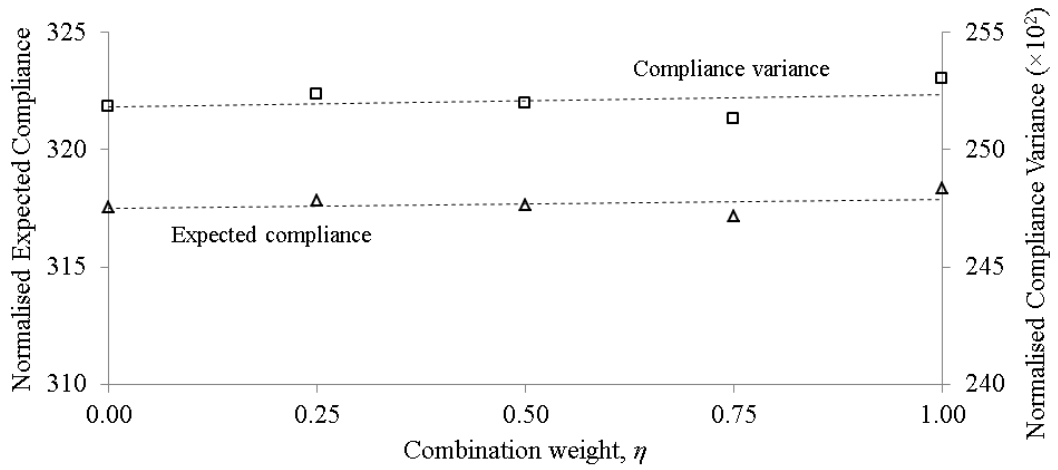

Figure 8-9: Bridge example. Expected and variance of compliance for a range of combination weights.

8.6 Conclusions

Variance is important in robust optimisation to reduce the sensitivity of performance when there are uncertainties in the operating environment. Therefore, compliance variance was introduced into the robust compliance minimisation problem. The analytical approach was used to derive a formulation for compliance variance under normally distributed loading magnitude uncertainties. This formulation allowed for analytical shape sensitivities to be derived using the adjoint method. Compliance variance was combined with expected compliance to formulate a combined robust objective. A non-dimensional weight based on the mean loading vector and Young's modulus was used to resolve the difference in units between the two components of the combined problem.

The analytical formulation for computing expected and variance of compliance was compared with a sampling method. The sampling method required more individual evaluations of compliance compared to the analytical formulation for an accurate computation. This effect became more pronounced for a greater number of uncertain variables. This highlights the importance of using an analytical approach to obtain an efficient method.

Results from two examples demonstrated that the effect of including compliance variance in robust optimisation is problem dependent. For the beam example there was a trade-off between expected and variance of compliance that depended on the weighting between the two parts of the objective. However, for the bridge example the two parts of the robust objective were mutual and the weighting had little effect. However, it is not clear why this occurs and what specific features of the two examples produce this contrasting behaviour.

Chapter 9

Conclusions

9.1 Concluding remarks

This section brings together the conclusions drawn throughout this thesis and links them back to the research objectives stated in Section 2.6. Structural topology optimisation allows the greatest freedom and hence greatest opportunity to find the best design. Level set based topology optimisation methods are becoming popular because they produce solutions with clear boundaries and are free from checkerboard patterns. The direct approach uses shape sensitivity analysis to derive a velocity function that moves the design towards an optimum. However, the direct method can be inefficient, dependent on the initial design and accurate sensitivity computation using a fixed grid FEA is a challenge. There are alternative level set methods that attempt to address one or more of these issues, although there does not appear to be a consensus on one particular approach. The direct approach is still attractive due to the availability of efficient and stable numerical schemes that were developed for other applications.

The analysis step is often the bottleneck in structural topology optimisation. Thus, an efficient FEA is critical for an efficient optimisation method. Hence, a

fixed grid FEM is usually employed in level set based optimisation to avoid time consuming re-meshing, although intersected elements are produced. The AFG method simply approximates intersected elements by weighting their stiffness by the area-fraction of real material within the element. This is a very efficient approach, but poor sensitivities computed on boundary can have a destabilising effect on the optimisation.

The first research objective was to investigate improve the sensitivity distribution obtained from the AFG method. The investigation is this work revealed sensitivity errors for the compliance objective increase as area-fraction decreases. This suggests the sensitivity distribution may not be smooth between neighbouring elements on the boundary, potentially leading to poor solutions and numerical instabilities. A weighted least squares method was proposed to improve boundary sensitivity computation when using the AFG method. Sensitivities were sampled at internal element points and weighted by area-faction divided by distance to the boundary. This approach was investigated and a set of suitable parameters were found that provided sufficient accuracy of the boundary sensitivity distribution.

The AFG method with the weighted least squares approach was sufficient to stably progress the design towards an optimum. However, small spurious fluctuations in boundary sensitivity near the optimum make it difficult to set an appropriate termination criterion. The second research objective was to create an alternative fixed grid method that can improve boundary sensitivity computation compared to the AFG method. A new fixed grid method was created by using elements that match the boundary exactly. This Boundary matching Fixed Grid (BFG) method is reasonably efficient because only those elements cut by the boundary require stiffness matrix computation each iteration. The BFG method showed potential for computing more accurate boundary sensitivities compared to the AFG method. However, artificial peaks in boundary sensitivity can be caused by spurious zero energy modes present in pentagonal elements.

The third objective was to implement an efficient and stable level set based structural optimisation method. A direct method was implemented to solve the minimisation of compliance problem. The Lagrange multiplier used to enforce the volume constraint was efficient computed each iteration using Newton's method. The implemented method was validated against classic examples. The results were consistent with those from other structural topology optimization methods.

Solutions obtained using the direct level set based topology optimisation method can be dependent on the initial design. This is mainly because there is no facility to create new holes during optimisation. Various methods have been proposed to enable new hole insertion. However, these methods either do not make a clear link between hole insertion and boundary propagation or can be sensitive to various parameters. The forth research objective was to create and test a hole insertion technique for the direct level method, that makes a meaningful link between boundary optimisation and hole insertion and is insensitive to the choice of parameters.

A new method was introduced in this thesis, where hole creation was enabled using a secondary implicit function that described an artificial height for a two dimensional structure. Hole creation was linked to boundary optimisation through the simultaneous update of the two implicit functions. The method was thoroughly investigated and found to be reasonably insensitive to parameters and converged fairly smoothly to good solutions of benchmark problems.

Uncertainty is important to consider during design and optimisation to produce robust and reliable designs. A robust optimisation problem was considered by minimising expected compliance under loading magnitude and direction uncertainty. When uncertainties are described by continuous functions, existing methods use discretization to approximate the robust problem as a multiple load case problem. However, this can become inefficient when there are a large number of uncertainties. Therefore, fifth research objective was to use an

analytical approach to efficiently solve the robust minimisation of expected compliance problem. The analytically derived formulation in this thesis showed that the number of load cases scales linearly with the number of uncertain loads. This is more efficient than existing methods that involve discretization.

Variability of performance under uncertainty is also important to consider in robust optimisation. Therefore the sixth research objective was to extend the robust compliance problem to include variance. A robust optimisation problem, including both expected and variance of compliance, was considered under loading magnitude uncertainty. The analytical approach was used to derive an efficient formulation for compliance variance and the adjoint method was used to compute shape sensitivities for the level set optimisation method. Examples demonstrated that the effect of including compliance variance in robust optimisation is problem dependent. For one example there was a trade-off between expected compliance and variance of compliance that depended on the weighting between the two parts of the objective. However, for a second example the two parts were mutual and the weighting had little effect.

9.2 Publications originating from this work

The work in this thesis has been published in the following journal papers:

- Dunning, P.D., Kim, H.A. & Mullineux, G., 2011. Investigation and improvement of sensitivity computation using the area-fraction weighted fixed grid FEM and structural optimization. *Finite Elements in Analysis and Design*, 47, pp. 933-941.
- Dunning, P.D., Kim, H.A. & Mullineux, G., 2011. Introducing Loading Uncertainty in Topology Optimization. *AIAA Journal*, 49(4), pp. 760-768.

The work in this thesis has also been presented internationally and published in the following conference papers:

- Dunning, P.D. & Kim, H.A., 2010c. A new hole insertion method for level set based topology optimization. *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, Texas, 13-15 Sept.
- Dunning, P.D., Kim, H.A. & Mullineux, G., 2010b. Introducing Uncertainty in Direction of Loading for Topology Optimization. *6th AIAA Multidisciplinary Design Optimization Specialist Conference*, Orlando, Florida, 12-15 April.
- Dunning, P.D., Kim, H.A. & Mullineux, G., 2010a. Loading Magnitude Uncertainty in Level-Set Based Topology Optimization. *12th AIAA Non-Deterministic Approaches Conference*, Orlando, Florida, 12-15 April.
- Dunning, P.D., Kim, H.A. & Mullineux, G., 2008b. Two-Dimensional Fixed Grid Based Finite Element Structural Analysis. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, Canada, 10-12 Sept.
- Dunning, P.D., Kim, H.A. & Mullineux, G., 2008a. Error Analysis of Fixed Grid Formulation for Boundary Based Structural Optimisation. *7th ASMO UK Conference on Engineering Design Optimization*, Bath, UK, 8-9 July.

The following papers are also in preparation or have been submitted:

- Dunning, P.D. & Kim, H.A., 2011. A new hole insertion method for level set based topology optimization. *International Journal for Numerical Methods in Engineering*, Submitted
- Dunning, P.D. & Kim, H.A., 2011. Robust topology optimization minimising expected compliance and variance. In preparation.

9.3 Future work

The BFG method showed potential for computing more accurate boundary sensitivities compared with the AFG method. However, the pentagonal element formulation has spurious zero energy modes that can lead to artificial peaks in sensitivity on the boundary. There are various techniques that could be employed to solve or eliminate this problem, including stabilising the element stiffness matrix or using an alternative pentagonal element formulation. These methods should be implemented and investigated for their suitability in the context of sensitivity computation for an iterative optimisation method.

The analytical approach to reformulating the minimisation of expected compliance problem employed in this thesis was limited to normally distributed uncertainties. Therefore, the approach should be extended to derive efficient formulations for other common probability distributions. When an analytical approach is not possible, then discretization techniques could be used and combined with the available analytical load cases to provide an efficient and flexible method.

The formulation and sensitivity analysis for compliance variance was limited to loading magnitude uncertainty described by normal distributions. This should be extended to include directional uncertainty and generalised to allow sensitivity computation for discrete probability distributions.

References

Abe, K., Kazama, S. & Koro, K., 2007. A boundary element approach for topology optimization problem using the level set method. *Communications In Numerical Methods In Engineering*, 23(5), pp.405-416.

Achtziger, W., 2007. On simultaneous optimization of truss geometry and topology. *Structural and Multidisciplinary Optimization*, 33(4-5), pp.285-304.

Adalsteinsson, D. & Sethian, J.A., 1995. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2), pp.269-277.

Adalsteinsson, D. & Sethian, J.A., 1999. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1), pp.2-22.

Allaire, G., Jouve, F. & Toader, A.M., 2004. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1), pp.363-393.

Allaire, G., de Gournay, F., Jouve, F., & Toader, A.M., 2005. Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics*, 34(1), pp.59-80.

Allaire, G. & Jouve, F., 2005. A level-set method for vibration and multiple loads structural optimization. *Computer Methods In Applied Mechanics and Engineering*, 194, pp.3269-3290.

Alvarez, F. & Carrasco, M., 2005. Minimization of the expected compliance as an alternative approach to multiload truss optimization. *Structural and Multidisciplinary Optimization*, 29(6), pp.470-476.

Amstutz, S. & Andra, H., 2006. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216(2), pp.573-588.

Arora, J.S., 2004. *Introduction to optimal design* 2nd ed., United States of America: Academic Press.

Baumgartner, A., Harzheim, L. & Mattheck, C., 1992. SKO (soft kill option) - the biological way to find an optimum structure topology. *International Journal of Fatigue*, 14(6), pp.387-393.

Belytschko, T. & Tsay, C.S., 1983. A stabilization procedure for the quadrilateral plate element with one-point quadrature. *International Journal For Numerical Methods In Engineering*, 19(3), pp.405-419.

Belytschko, T., Moes, N., Usui, S. & Parimi, C., 2001. Arbitrary discontinuities in finite elements. *International Journal For Numerical Methods In Engineering*, 50, pp.993-1013.

Belytschko, T., Parimi, C., Moes, N., Sukumar, N. & Usui, S, 2003a. Structured extended finite element methods for solids defined by implicit surfaces. *International Journal For Numerical Methods In Engineering*, 56(4), pp.609-635.

Belytschko, T., Xiao, S.P. & Parimi, C., 2003b. Topology optimization with implicit functions and regularization. *International Journal For Numerical Methods In Engineering*, 57, pp.1177-1196.

Ben-Tal, A. & Nemirovski, A., 1997. Robust truss topology design via semidefinite programming. *Siam Journal On Optimization*, 7(4), pp.991-1016.

Bendsøe, M.P., 1989. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4), pp.193-202.

Bendsøe, M.P. & Kikuchi, N., 1988. Generating optimal topologies in structural design using a homogenization method. *Computer Methods In Applied Mechanics and Engineering*, 71(2), pp.197-224.

Bendsøe, M.P. & Sigmund, O., 2004. *Topology optimization: Theory, Methods and Applications*, Germany: Springer-Verlag.

Boggs, P.T. & Tolle, J.W., 1995. Sequential quadratic programming. *Acta Numerica*, 4, pp.1-51.

Broyden, C.G., 1967. Quasi-Newton methods and their application to function minimization. *Mathematics of Computation*, 21, pp.368-381.

Bulman, S., Sienz, J. & Hinton, E., 2001. Comparisons between algorithms for structural topology optimization using a series of benchmark studies. *Computers and Structures*, 79(12), pp.1203-1218.

Burger, M., Hackl, B. & Ring, W., 2004. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194(1), pp.344-362.

Calafiore, G.C. & Dabbene, F., 2008. Optimization under uncertainty with applications to design of truss structures. *Structural and Multidisciplinary Optimization*, 35(3), pp.189-200.

Céa, J., Garreau, S., Guillaume, P. & Masmoudi, M., 2000. The shape and topological optimizations connection. *Computer Methods In Applied Mechanics and Engineering*, 188(4), pp.713-726.

Cervera, E. & Trevelyan, J., 2005. Evolutionary structural optimisation based on boundary representation of NURBS. Part I: 2D algorithms. *Computers and Structures*, 83(23-24), pp.1902-1916.

Challis, V.J., 2010. A discrete level-set topology chooptimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 41(3), pp.453-464.

Chen, S., Chen, W. & Lee, S.H., 2010. Level set based robust shape and topology optimization under random field uncertainties. *Structural and Multidisciplinary Optimization*, 41(4), pp.507-524.

Choi, S., Grandhi, R.V. & Canfield, R.A., 2007. *Reliability-based Structural Design*, London: Springer.

Chopp, D.L., 1993. Computing Minimal Surfaces via Level Set Curvature Flow. *Journal of Computational Physics*, 106(1), pp.77-91.

Christiansen, S., Patriksson, M. & Wynter, L., 2001. Stochastic bilevel programming in structural optimization. *Structural and Multidisciplinary Optimization*, 21(5), pp.361-371.

Conti, S., Held, H., Pach, M., Rumpf, M., & Schultz, R., 2008. Shape optimization under uncertainty-a stochastic programming perspective. *SIAM Journal On Optimization*, 19(4), pp.1610-1632.

Cook, R.D., Malkus, D.S., Plesha, M.E., & Witt, R.J., 2002. *Concepts and applications of finite element analysis* 4th ed., United States: John Wiley & Sons, Inc.

Dantzig, G.B. & Thapa, M.N., 1997. *Linear Programming: Introduction*, United States of America: Springer.

Dantzig, G.B. & Wolfe, P., 1960. Decomposition principle for linear programs. *Operations research*, 8, pp.101-111.

Daux, C., Moes, N., Dolbow, J., Sukumar, N. & Belytschko, T., 2000. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal For Numerical Methods In Engineering*, 48, pp.1741-1760.

Diaz, A. & Sigmund, O., 1995. Checkerboard patterns in layout optimization. *Structural Optimization*, 10(1), pp.40-45.

Doltsinis, I. & Kang, Z., 2004. Robust design of structures using optimization methods. *Computer Methods In Applied Mechanics and Engineering*, 193(23-26), pp.2221-2237.

Duff, I.S., 2002. MA57 - A new code for the solution of sparse symmetric definite and indefinite systems. Technical report RAL-TR-2002-024, Rutherford Appleton Laboratory, Oxon.

Edwards, C.S., 2008. *The analysis and development of efficient and reliable topology optimisation schemes*. Thesis. UK: University of Bath.

Eschenauer, H.A., Kobelev, V.V. & Schumacher, A., 1994. Bubble method for topology and shape optimization of structures. *Structural Optimization*, 8(1), pp.42-51.

Evgrafov, A. & Patriksson, M., 2003. Stochastic structural topology optimization: discretization and penalty function approach. *Structural and Multidisciplinary Optimization*, 25(3), pp.174-188.

Fletcher, R. & Reeves, C.M., 1964. Function minimization by conjugate gradients. *Computer Journal*, 7(2), pp.149-154.

Garcia-Ruiz, M.J. & Steven, G.P., 1999. Fixed grid finite elements in elasticity problems. *Engineering Computations*, 16(2), pp.145-164.

Gen, M. & Cheng, R., 1997. *Genetic algorithms and engineering design*, United States of America: John Wiley & Sons.

Gomes, A.A. & Suleman, A., 2006. Application of spectral level set methodology in topology optimization. *Structural and Multidisciplinary Optimization*, 31(6), pp.430-443.

Gomes, A.A. & Suleman, A., 2008. Topology optimization of a reinforced wing box for enhanced roll maneuvers. *AIAA Journal*, 46(3), pp.548-556.

Gournay, F.D., Allaire, G. & Jouve, F., 2008. Shape and topology optimization of the robust compliance via the level set method. *ESAIM: Control, Optimisation and Calculus of Variations*, 14(1), pp.43-70.

Guest, J.K. & Igusa, T., 2008. Structural optimization under uncertain loads and nodal locations. *Computer Methods In Applied Mechanics and Engineering*, 198(1), pp.116-124.

Ha, S. & Cho, S., 2008. Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Computers and Structures*, 86(13-14), pp.1447-1455.

Haber, R.B., Jog, C.S. & Bendsøe, M.P., 1996. A new approach to variable-topology shape design using a constraint on perimeter. Structural Optimization, 11(1), pp.1-12.

Haftka, R.T. & Gürdal, Z., 1992. *Elements of Structural Optimization* 3rd ed., London: Kluwer Academic.

He, L., Kao, C. & Osher, S., 2007. Incorporating topological derivatives into shape derivatives based level set methods. *Journal of Computational Physics*, 225(1), pp.891-909.

Hinton E. & Sienz J., 1995. Fully stressed topological design of structures using an evolutionary procedure. *Engineering Computations*, 12(3), pp.229-244.

Hsu, Y.L., Hsu, M.S. & Chen, C.T., 2001. Interpreting results from topology optimization using density contours. *Computers and Structures*, 79(10), pp.1049-1058.

Huang, X., Xie, Y.M. & Burry, M.C., 2007. Advantages of bi-directional evolutionary structural optimization (BESO) over evolutionary structural optimization (ESO). *Advances In Structural Engineering*, 10(6), pp.727-737.

Jang, G. & Kim, Y.Y., 2005. Sensitivity analysis for fixed-grid shape optimization by using oblique boundary curve approximation. *International Journal of Solids and Structures*, 42(11-12), pp.3591-3609.

Jang, G., Kim, Y.Y. & Choi, K.K., 2004. Remesh-free shape optimization using the wavelet-Galerkin method. *International Journal of Solids and Structures*, 41(22-23), pp.6465-6483.

Jiang, G. & Peng, D., 2000. Weighted ENO schemes for Hamilton-Jacobi equations. *Siam Journal On Scientific Computing*, 21(6), pp.2126-2143.

Jog, C.S. & Haber, R.B., 1996. Stability of finite element models for distributed-parameter optimization and topology design. *Computer Methods In Applied Mechanics and Engineering*, 130(3-4), pp.203-226.

Jung, H.S. & Cho, S., 2004. Reliability-based topology optimization of geometrically nonlinear structures with loading and material uncertainties. *Finite Elements In Analysis and Design*, 43(3), pp.311-331.

Kang, Z. & Luo, Y., 2009. Non-probabilistic reliability-based topology optimization of geometrically nonlinear structures using convex models. *Computer Methods In Applied Mechanics and Engineering*, 198(41-44), pp.3228-3238.

Kang, Z. & Luo, Y., 2010. Reliability-based structural optimization with probability and convex set hybrid models. *Structural and Multidisciplinary Optimization*, 42(1), pp.89-102.

Kharmanda, G., Olhoff, N., Mohamed, A., & Lemaire, M., 2004. Reliability-based topology optimization. *Structural and Multidisciplinary Optimization*, 26(5), pp.295-307.

Karmarkar, N., 1984. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4(4), pp.373-395.

Kelley, J.E., 1960. The Cutting-Plane Method for Solving Convex Programs. *Journal of SIAM*, 8(4), pp.703-712.

Kim, C., Wang, S.Y., Bae, K.R., Moon, H., & Choi, K.K., 2006. Reliability-based topology optimization with uncertainties. *Journal of Mechanical Science and Technology*, 20(4), pp.494-504.

Kim, H., Garcia, M.J., Querin, O.M., Steven, G.P. & Xie, Y.M., 2000. Introduction of fixed grid in evolutionary structural optimisation. *Engineering Computations*, 17(4), pp.427-439.

Kim, M., Ha, S. & Cho, S., 2009. Level Set-Based Topological Shape Optimization of Nonlinear Heat Conduction Problems Using Topological Derivatives. *Mechanics Based Design of Structures and Machines*, 37(4), pp.550-582.

Kim, N.H. & Chang, Y.M., 2005. Eulerian shape design sensitivity analysis and optimization with a fixed grid. *Computer Methods In Applied Mechanics and Engineering*, 194, pp.3291-3314.

Kocvara, M., Zowe, J. & Nemirovski, A., 2000. Cascading - an approach to robust material optimization. *Computers & Structures*, 76(1-3), pp.431-442.

Kogiso, N., Ahn, W., Nishiwaki, S., Izui, K., & Yoshimura, M., 2008. Robust Topology Optimization for Compliant Mechanisms Considering Uncertainty of Applied Loads. Journal of Advanced Mechanical Design Systems and Manufacturing, 2(1), pp.96-107.

Laarhoven, P.J.M. & Aarts, E.H.L, 1987. *Simulated annealing: theory and applications*, Netherlands: Kluwer Academic.

Lee, S.H., Chen, W. & Kwak, B.M., 2009. Robust design with arbitrary distributions using Gauss-type quadrature formula. *Structural and Multidisciplinary Optimization*, 39(3), pp.227-243.

Lee, S. & Kwak, B.M., 2008. Smooth boundary topology optimization for eigenvalue performance and its application to the design of a flexural stage. *Engineering Optimization*, 40(3), pp.271-285.

Lee, S., Kwak, B.M. & Kim, I.Y., 2007. Smooth Boundary Topology Optimization Using B-spline and Hole Generation. *International Journal of CAD/CAM*, 7(1).

Li, Q., Steven, G.P. & Xie, Y.M., 2001. A simple checkerboard suppression algorithm for evolutionary structural optimization. *Structural and Multidisciplinary Optimization*, 22, pp.230-239.

Liu, G., 2003. *Mesh free methods: Moving beyond the finite element method*, United States of America: CRC Press LLC.

Liu, Z., Korvink, J.G. & Huang, R., 2005. Structure topology optimization: fully coupled level set method via FEMLAB. *Structural and Multidisciplinary Optimization*, 29(6), pp.407-417.

Liu, Z. & Korvink, J.G., 2008. Adaptive moving mesh level set method for structure topology optimization. *Engineering Optimization*, 40(6), pp.529-558.

Logo, J., 2007. New type of optimality criteria method in case of probabilistic loading conditions. *Mechanics Based Design of Structures and Machines*, 35(2), pp.147-162.

Luo, J., Luo, Z., Chen, L., Tong, L. & Wang, M.Y., 2008. A semi-implicit level set method for structural shape and topology optimization. *Journal of Computational Physics*, 227(11), pp.5561-5581.

Luo, Y., Kang, Z., Luo, Z. & Li., A., 2009. Continuum topology optimization with non-probabilistic reliability constraints based on multi-ellipsoid convex model. *Structural and Multidisciplinary Optimization*, 39(3), pp.297-310.

Luo, Z., Tong, L., Wang, M.Y. & Wang, S.Y., 2007. Shape and topology optimization of compliant mechanisms using a parameterization level set method. *Journal of Computational Physics*, 227(1), pp.680-705.

Luo, Z. & Tong, L., 2008. A level set method for shape and topology optimization of large-displacement compliant mechanisms. *International Journal For Numerical Methods In Engineering*, 76(6), pp.862-892.

Luo, Z., Tong, L. & Wang, M.Y., 2008a. Design of distributed compliant micromechanisms with an implicit free boundary representation. *Structural and Multidisciplinary Optimization*, 36(6), pp.607-621.

Luo, Z., Wang, M.Y., Wang, S.Y. & Wei, P., 2008b. A level set-based parameterization method for structural shape and topology optimization. *International Journal For Numerical Methods In Engineering*, 76(1), pp.1-26.

Luo, Z., Tong, L. & Kang, Z., 2009a. A level set method for structural shape and topology optimization using radial basis functions. *Computers and Structures*, 87, pp.425-434.

Luo, Z., Tong, L. Luo, J., Wei, P. & Wang, M.Y., 2009b. Design of piezoelectric actuators using a multiphase level set method of piecewise constants. *Journal of Computational Physics*, 228(7), pp.2643-2659.

Maute, K. & Frangopol, D.M., 2003. Reliability-based design of MEMS mechanisms by topology optimization. *Computers and Structures*, 81(8-11), pp.813-824.

Mei, Y.L. & Wang, X., 2004. A level set method for structural topology optimization and its applications. *Advances In Engineering Software*, 35, pp.415-441.

Michell, A.G., 1904. Limits of Economy of Material in Frame-structures. *Philosophical Magazine*, 8(47), pp.589-597.

Mogami, K., Nishiwaki, S., Izui, K., Yoshimura, M. & Kogiso, N., 2006. Reliability-based structural optimization of frame structures for multiple failure criteria using topology optimization techniques. *Structural and Multidisciplinary Optimization*, 32(4), pp.299-311.

Norato, J., Haber, R., Tortorelli, D. & Bendsøe, M.P., 2004. A geometry projection method for shape optimization. *International Journal For Numerical Methods In Engineering*, 60, pp.2289-2312.

Norato, J.A., Bendsøe, M.P., Haber, R.B. & Tortorelli, D.A., 2007. A topological derivative method for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5), pp.375-386.

Osher, S.J. & Fedkiw, R., 2003. *Level set methods and dynamic implicit surfaces*, New York: Springer.

Osher, S.J. & Santosa, F., 2001. Level set methods for optimization problems involving geometry and constraints I. Frequencies of a two-density inhomogeneous drum. *Journal of Computational Physics*, 171(1), pp.272-288.

Park, G-J., Lee, T-H., Lee, K.H. & Hwang, K-H., 2006. Robust design: An overview. *AIAA Journal*, 44(1), pp.181-191.

Park, K. & Youn, S., 2008. Topology optimization of shell structures using adaptive inner-front (AIF) level set method. *Structural and Multidisciplinary Optimization*, 36(1), pp.43-58.

Peters, J.F. & Heymsfield, E., 2003. Application of the 2-D constant strain assumption to FEM elements consisting of an arbitrary number of nodes. *International Journal of Solids and Structures*, 40(1), pp.143-159.

Peters, J.F. & Heymsfield, E., 2004. FEM formulation of four and five noded elements using a linearly varying stress assumption. *International Journal of Solids and Structures*, 41(7), pp.1991-2009.

Petersson, J. & Sigmund, O., 1998. Slope constrained topology optimization. *International Journal For Numerical Methods In Engineering*, 41(8), pp.1417-1434.

Powell, M.J.D., 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(4), pp.303-307.

Querin, O.M., Young, V., Steven, G.P. & Xie, Y.M., 2000. Computational efficiency and validation of bi-directional evolutionary structural optimisation. Computer Methods In Applied Mechanics and Engineering, 189(2), pp.559-573.

Rao, S.S., 2009. Engineering optimization: theory and practice 4th ed., United States of America: John Wiley & Sons.

Rashid, M.M. & Gullett, P.M., 2000. On a finite element method with variable element topology. *Computer Methods In Applied Mechanics and Engineering*, 190(11-12), pp.1509-1527.

Rong, J. & Liang, Q., 2008. A level set method for topology optimization of continuum structures with bounded design domains. *Computer Methods In Applied Mechanics and Engineering*, 197(17), pp.1447-1465.

Rossow, M.P. & Taylor, J.E., 1973. Finite-element method for optimal design of variable thickness sheets. *AIAA Journal*, 11(11), pp.1566-1569.

Rozvany, G.I., 2001. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21(2), pp.90-108.

Sethian, J.A., 1999. *Level set methods and fast marching methods* 2nd ed., New York: Cambridge University Press.

Sethian, J.A. & Wiegmann, A., 2000. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163, pp.489-528.

Sigmund, O. & Petersson, J., 1998. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1), pp.68-75.

Silva, M., Tortorelli, D.A., Norato, J.A., Ha, C. & Bae, H-R., 2010. Component and system reliability-based topology optimization using a single-loop method. *Structural and Multidisciplinary Optimization*, 41(1), pp.87-106.

Strang, G. & Kohn, R.V., 1986. Optimal-design in elasticity and placticity. *International Journal For Numerical Methods In Engineering*, 22(1), pp.183-188.

Suzuki, K. & Kikuchi, N., 1991. A homogenization method for shape and topology optimization. *Computer Methods In Applied Mechanics and Engineering*, 93(3), pp.291-318.

Takezawa, A., Nishiwaki, S. & Kitamura, M., 2010. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 229(7), pp.2697-2718.

Tang, P.S. & Chang, K.H., 2001. Integration of topology and shape optimization for design of structural components. *Structural and Multidisciplinary Optimization*, 22(1), pp.65-82.

Van Miegroet, L. & Duysinx, P., 2007. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization*, 33(4), pp.425-438.

Wang, M.Y., Wang, X. & Guo, D., 2003. A level set method for structural topology optimization. *Computer Methods In Applied Mechanics and Engineering*, 192(1-2), pp.227-246.

Wang, S.Y. & Wang, M.Y., 2006a. A moving superimposed finite element method for structural topology optimization. *International Journal For Numerical Methods In Engineering*, 65, pp.1892-1922.

Wang, S.Y. & Wang, M.Y., 2006b. Radial basis functions and level set method for structural topology optimization. *International Journal For Numerical Methods In Engineering*, 65(12), pp.2060-2090.

Wang, S.Y., Lim, K.M., Khoo, B.C. & Wang, M.Y., 2007. An extended level set method for shape and topology optimization. *Journal of Computational Physics*, 221(1), pp.395-421.

Wang, X., Wang, M.Y. & Guo, D., 2004. Structural shape and topology optimization in a level-set-based framework of region representation. *Structural and Multidisciplinary Optimization*, 27, pp.1-19.

Wei, P. & Wang, M.Y., 2009. Piecewise constant level set method for structural topology optimization. *International Journal For Numerical Methods In Engineering*, 78(4), pp.379-402.

Wei, P., Wang, M.Y. & Xing, X., 2010. A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design*, 42(8), pp.708-719.

Xia, Q., Wang, M.Y., Wang, S.Y. & Chen, S., 2006. Semi-Lagrange method for level-set-based structural topology and shape optimization. *Structural and Multidisciplinary Optimization*, 31(6), pp.419-429.

Xie, Y.M. & Steven, G.P., 1997. *Evolutionary structural optimization*, Springer.

Zangwill, W.I., 1967. Non-linear programming via penalty functions. *Management science*, 13(5), pp.344-358.

Zhang, X. & Ouyang, G., 2008. A level set method for reliability-based topology optimization of compliant mechanisms. *Science In China Series E-Technological Sciences*, 51(4), pp.443-455.

Zhuang, C., Xiong, Z. & Ding, H., 2007. A level set method for topology optimization of heat conduction problem under multiple load cases. Computer Methods In Applied Mechanics and Engineering, 196(4-6), pp.1074-1084.

Appendix 1

Derivation of robust optimisation formulations

A1.1 Introduction

The purpose of this appendix is to provide further details for the derivation of the robust optimisation formulations used in Chapters 7 and 8. The appendix has two sections, the first is for expected compliance under loading magnitude and direction uncertainty. The second section details the derivation for compliance variance under loading magnitude uncertainty.

A1.1.1 Useful equations

This section presents some useful equations used for the derivations detailed in this appendix. First, an uncertain load magnitude is normalised using its mean and standard deviation:

$$\bar{f}_i = (f_i - \mu_i)/\sqrt{2}\sigma_i \tag{A1.1}$$

The normal Probability Density Function (PDF) for the uncertain load magnitude can be defined as:

$$P(f_i) = \exp(-\bar{f}_i^2)/\sqrt{(2\pi\sigma_i^2)} \tag{A1.2}$$

The integral of a normal PDF involves the error function that is defined as:

$$erf(\bar{f}_i) = (2/\sqrt{\pi})\int_0^{\bar{f}_i} \exp(-t^2)\,dt \tag{A1.3}$$

The first and subsequent integrals of the normal PDF can be derived as:

$$\int P(f_i)\,df_i = 1/2\,erf\left(\bar{f}_i\right) \tag{A1.4}$$

$$\int\int P(f_i)\,df_i = \left(\bar{f}_i\sigma_i/\sqrt{2}\right)\left(erf(\bar{f}_i)\right) + \sigma_i^2 P(f_i) \tag{A1.5}$$

$$\int\int\int P(f_i)\,df_i = \left(\frac{\sigma_i^2}{2}\right)\left[\left(\frac{1}{2} + \bar{f}_i^2\right)erf\left(\bar{f}_i\right)\right] + \left(\bar{f}_i\sigma_i^3/2\right)P(f_i) \tag{A1.6}$$

$$\int\int\int\int P(f_i)\,df_i = \left(\frac{\sigma_i^3}{\sqrt{2}}\right)\left[\left(\frac{\bar{f}_i}{2} + \frac{\bar{f}_i^3}{3}\right)erf\left(\bar{f}_i\right)\right] + \left(\frac{\sigma_i^4}{3}\right)(1 + \bar{f}_i^2)\,P(f_i) \tag{A1.7}$$

$$\begin{aligned}\int\int\int\int\int P(f_i)\,df_i = &\left(\frac{\sigma_i^4}{2}\right)\left[\left(\frac{1}{8} + \frac{\bar{f}_i^2}{2} + \frac{\bar{f}_i^4}{6}\right)erf\left(\bar{f}_i\right)\right] \\ &+ \left(\frac{5\,\bar{f}_i + 2\,\bar{f}_i^3}{12\sqrt{2}}\right)\sigma_i^5 P(f_i)\end{aligned} \tag{A1.8}$$

A1.2 Expected compliance

A1.2.1 Loading magnitude uncertainty

This section evaluates the term in the square brackets in (7.5) between limits of $\mu_1 \pm \zeta$. Starting with:

$$\begin{aligned}&\left(\kappa_{1,1} f_1^2 + 2 f_1 \sum_{i=2}^{m} \kappa_{1,i} f_i + \sum_{i,\,j=2}^{m} \kappa_{i,\,j} f_i f_j\right) \int P(f_1)\, df_1 \\ &-2\left(\kappa_{1,1} f_1 + \sum_{i=2}^{m} \kappa_{1,i} f_i\right) \int\int P(f_1)\, df_1 + 2\kappa_{1,1} \int\int\int P(f_1)\, df_1\end{aligned} \tag{A1.9}$$

then substituting (A1.4) to (A1.6) into (A1.9) and collecting terms yields:

$$\begin{aligned}&1/2\, erf(\bar{f}_1)\left(\kappa_{1,1}(\mu_1^2 + \sigma_1^2) + 2\mu_1 \sum_{i=2}^{m} \kappa_{1,i} f_i + \sum_{i,\,j=2}^{m} \kappa_{i,\,j} f_i f_j\right) \\ &+\sigma_1^2 P(f_1)\left(\kappa_{1,1}(\sigma_1 \bar{f}_1 - 2 f_1) - 2 f_1 \sum_{i=2}^{m} \kappa_{1,i} f_i\right)\end{aligned} \tag{A1.10}$$

Evaluating (A1.10) between limits, $f_1 = \mu_1 \pm \zeta$, then letting $\zeta \rightarrow \infty$ and using the properties that $erf(\infty) = 1$, $erf(-\infty) = -1$ and $P(\pm\infty) = 0$, yields:

$$\kappa_{1,1}(\mu_1^2 + \sigma_1^2) + 2\mu_1 \sum_{i=2}^{m} \kappa_{1,i} f_i + \sum_{i,\,j=2}^{m} \kappa_{i,\,j} f_i f_j \tag{A1.11}$$

which is the term in the square brackets for equation (7.6).

A1.2.2 Loading direction uncertainty

This section provides further detail of the steps required to get to (7.12) using (7.9) and (7.11). First the complex exponential forms of the trigonometric functions are substituted into (7.11):

$$\begin{aligned} C(f,\theta) = \sum_{i,j=1}^{ml/2} \left(\frac{f_i f_j}{4\exp(I\theta_i)\exp(I\theta_j)} \right) \\ \times \left(\kappa_{ix,jx}(\exp(2I\theta_i)+1)(\exp(2I\theta_j)+1) \right. \\ -\kappa_{iy,jy}(\exp(2I\theta_i)-1)(\exp(2I\theta_j)-1) \\ \left. -2\kappa_{ix,jy}(\exp(2I\theta_i)+1)(\exp(2I\theta_j)-1) \right) \end{aligned} \tag{A1.12}$$

where I = √-1. Substituting (A1.12) into (7.9) gives a formulation that can be directly integrated. Evaluating the integral for the first uncertain direction, θ_1, between limits $\theta_1=\mu_{\theta,1}\pm\pi$, which integrates over a full revolution of 2π, produces:

$$\begin{aligned} E[C] = erf(\pi/\sqrt{2}\sigma_{\theta 1}) \left[(1/2) \left[\kappa_{1x,1x} \left(1+\exp(-2\sigma_{\theta 1}^2)\cos(2\mu_{\theta 1}) \right) \right. \right. \\ \left. \left. +\kappa_{1y,1y} \left(1-\exp(-2\sigma_{\theta 1}^2)\cos(2\mu_{\theta 1}) \right) + \kappa_{1x,1y} \left(\exp(-2\sigma_{\theta 1}^2)\sin(2\mu_{\theta 1}) \right) \right] \right. \\ +2f_1\exp(-\sigma_{\theta 1}^2/2) \int_{\theta_2} \cdots \int_{\theta_n} \sum_{i=2}^{ml/2} f_i \left[\kappa_{1x,ix}\cos(\mu_{\theta 1})\cos(\theta_i) \right. \\ +\kappa_{1y,iy}\sin(\mu_{\theta 1})\sin(\theta_i) + \kappa_{1x,iy}\cos(\mu_{\theta 1})\sin(\theta_i) \\ \left. +\kappa_{ix,1y}\sin(\mu_{\theta 1})\cos(\theta_i) \right] \prod_{i=2}^{n} P(\theta_i)\, df_2 \cdots df_n \\ +\int_{\theta_2} \cdots \int_{\theta_n} \sum_{i,j=1}^{ml/2} f_i f_j \left(\kappa_{ix,jx}\cos(\theta_i)\cos(\theta_j) + \kappa_{iy,jy}\sin(\theta_i)\sin(\theta_j) \right. \\ \left. \left. +2\kappa_{ix,jy}\cos(\theta_i)\sin(\theta_j) \right) \prod_{i=2}^{n} P(\theta_i)\, df_2 \cdots df_n \right] \end{aligned} \tag{A1.13}$$

Now if, $3\sigma_{\theta,1} \le \pi$, then, $1.0 > \text{erf}[\pi/(\sigma_{\theta,i}\sqrt{2})] > 0.9973$, which can be approximated as one. Using this approximation and repeating the integration over all loads with uncertain direction gives the formulation for expected compliance (7.12).

A1.3 Compliance variance

This section details additional steps between (8.4) and (8.5). First square (8.3):

$$C(f)^2 = \sum_{i,j,p,q=1}^{m/2} f_i f_j f_p f_q \Theta_1 \tag{A1.14}$$

where Θ_1 is defined by (8.6). Substituting (A1.14) into (8.4), the integral is evaluated for the first uncertain load, f_1, using integration by parts five times:

$$\begin{aligned} E[C(f)^2] = \int_{f_2}\cdots\int_{f_n} \Big[& \left(\bar{\Theta}_1 f_1^4 + \bar{\Theta}_2 f_1^3 + \bar{\Theta}_3 f_1^2 + \bar{\Theta}_4 f_1 + \bar{\Theta}_5\right)\int P(f_1)\,df_1 \\ & - \left(4\bar{\Theta}_1 f_1^3 + 3\bar{\Theta}_2 f_1^2 + 2\bar{\Theta}_3 f_1 + \bar{\Theta}_4\right)\int\int P(f_1)\,df_1 \\ & + \left(12\bar{\Theta}_1 f_1^2 + 6\bar{\Theta}_2 f_1 + 2\bar{\Theta}_3\right)\int\int\int P(f_1)\,df_1 \\ & - \left(24\bar{\Theta}_1 f_1 + 6\bar{\Theta}_2\right)\int\int\int\int P(f_1)\,df_1 \\ & + \left(24\bar{\Theta}_1\right)\int\int\int\int\int P(f_1)\,df_1 \Big] \prod_{i=2}^{n} P(f_i)\,df_2\cdots df_n \end{aligned} \tag{A1.15}$$

where the terms $\bar{\Theta}_k$ are defined as:

$$\begin{aligned} \bar{\Theta}_1 = {} & \kappa_{1x,1x}^2 \cos^4(\theta_1) + \kappa_{1y,1y}^2 \sin^4(\theta_1) \\ & + (4\kappa_{1x,1y}^2 + 2\kappa_{1x,1x}\kappa_{1y,1y})\cos^2(\theta_1)\sin^2(\theta_1) \\ & + 4\kappa_{1x,ix}\kappa_{1x,1y}\cos^3(\theta_1)\sin(\theta_1) \\ & + 4\kappa_{1y,1y}\kappa_{1x,1y}\sin^3(\theta_1)\cos(\theta_1) \end{aligned} \tag{A1.16}$$

$$\begin{aligned} \bar{\Theta}_2 = 2\sum_{i=2}^{m/2} f_i \Big[& \kappa_{1x,1x}\kappa_{1x,ix}\cos^3(\theta_1)\cos(\theta_i) + \kappa_{1y,1y}\kappa_{1y,iy}\sin^3(\theta_1)\sin(\theta_i) \\ & + (2\kappa_{1x,1y}\kappa_{1x,iy} + \kappa_{1x,1x}\kappa_{1y,iy})\cos^2(\theta_1)\sin(\theta_1)\sin(\theta_i) \\ & + (2\kappa_{1x,1y}\kappa_{ix,1y} + \kappa_{1x,ix}\kappa_{1y,1y})\sin^2(\theta_1)\cos(\theta_1)\cos(\theta_i) \\ & + \kappa_{1x,1x}\kappa_{1x,iy}\cos^3(\theta_1)\sin(\theta_i) + \kappa_{1y,1y}\kappa_{ix,1y}\sin^3(\theta_1)\cos(\theta_i) \\ & + (2\kappa_{1x,ix}\kappa_{1x,1y} + \kappa_{1x,1x}\kappa_{ix,1y})\cos^2(\theta_1)\sin(\theta_1)\cos(\theta_i) \\ & + (2\kappa_{1y,iy}\kappa_{1x,1y} + \kappa_{1y,1y}\kappa_{1x,iy})\sin^2(\theta_1)\cos(\theta_1)\sin(\theta_i) \Big] \end{aligned} \tag{A1.17}$$

$$\bar{\Theta}_3 = \sum_{i,j=2}^{m/2} f_i f_j \left[4\Theta_{2,p=1} + 2\Theta_{3,p=1}\right] \tag{A1.18}$$

$$\begin{aligned}\bar{\Theta}_4 = 4 \sum_{i,j,p=2}^{m/2} f_i f_j f_p \big[& \kappa_{1x,ix}\kappa_{jx,px}\cos(\theta_1)\cos(\theta_i)\cos(\theta_j)\cos(\theta_p) \\ & \kappa_{1y,iy}\kappa_{jy,py}\sin(\theta_1)\sin(\theta_i)\sin(\theta_j)\sin(\theta_p) \\ & +(\kappa_{1x,jy}\kappa_{ix,py}+\kappa_{1x,ix}\kappa_{jy,py})\cos(\theta_1)\cos(\theta_i)\sin(\theta_j)\sin(\theta_p) \\ & +(\kappa_{jx,1y}\kappa_{px,iy}+\kappa_{jx,px}\kappa_{1y,iy})\sin(\theta_1)\sin(\theta_i)\cos(\theta_j)\cos(\theta_p) \\ & +\kappa_{1x,ix}\kappa_{jx,py}\cos(\theta_1)\cos(\theta_i)\cos(\theta_j)\sin(\theta_p) \\ & +\kappa_{1y,iy}\kappa_{px,jy}\sin(\theta_1)\sin(\theta_i)\sin(\theta_j)\cos(\theta_p)\big]\end{aligned} \tag{A1.19}$$

$$\bar{\Theta}_5 = \sum_{i,j,p,q=2}^{m/2} f_i f_j f_p f_q \Theta_1 \tag{A1.20}$$

Substituting (A1.4) to (A1.8) into (A1.15) and rearranging, the term in the square brackets for (A1.15) becomes:

$$\begin{aligned} & 1/2\, erf(\bar{f}_1)\Big[\bar{\Theta}_1\left((\mu_1^2+\sigma_1^2)^2+2\sigma_1^4+4\mu_1^2\sigma_1^2\right)+\bar{\Theta}_2\left(\mu_1^3+3\mu_1\sigma_1^2\right) \\ & \qquad +\bar{\Theta}_3\left(\mu_1^2+\sigma_1^2\right)+\bar{\Theta}_4\mu_1+\bar{\Theta}_5\Big] \\ & +P(f_1)\Big[\bar{\Theta}_1\left(\sqrt{2}\sigma_1^5(5\bar{f}_1+2\bar{f}_1^3)-8f_1\sigma_1^4(1+\bar{f}_1)+6f_i^2\sigma_1^3\bar{f}_1-4f_1^3\sigma_1^2\right) \\ & \qquad +\bar{\Theta}_2\left(3f_1\sigma_1^3\bar{f}_1-3f_1^2\sigma_1^2\right)+\bar{\Theta}_3\left(\sigma_1^3\bar{f}_1-2f_1\sigma_1^2\right)-\bar{\Theta}_4\sigma_1^2\Big]\end{aligned} \tag{A1.21}$$

Evaluating (A1.21) between limits, $f_1 = \mu_1 \pm \zeta$, then letting $\zeta \rightarrow \infty$ and using the properties that $erf(\infty) = 1$, $erf(-\infty) = -1$ and $P(\pm\infty) = 0$, yields:

$$\begin{aligned} & \bar{\Theta}_1\left((\mu_1^2+\sigma_1^2)^2+2\sigma_1^4+4\mu_1^2\sigma_1^2\right)+\bar{\Theta}_2\left(\mu_1^3+3\mu_1\sigma_1^2\right) \\ & \qquad +\bar{\Theta}_3\left(\mu_1^2+\sigma_1^2\right)+\bar{\Theta}_4\mu_1+\bar{\Theta}_5\end{aligned} \tag{A1.22}$$

Substituting (A1.22) back into (A1.15) gives:

$$\begin{aligned} E[C(f)^2] = \int_{f_2}\cdots\int_{f_n}\Big[& \bar{\Theta}_1\left((\mu_1^2+\sigma_1^2)^2+2\sigma_1^4+4\mu_1^2\sigma_1^2\right)+\bar{\Theta}_2\left(\mu_1^3+3\mu_1\sigma_1^2\right) \\ & +\bar{\Theta}_3\left(\mu_1^2+\sigma_1^2\right)+\bar{\Theta}_4\mu_1+\bar{\Theta}_5\Big]\prod_{i=2}^{n}P(f_i)\,df_2\cdots df_n\end{aligned} \tag{A1.23}$$

Following the same integration process for the remaining uncertain loads and combining terms yields the final analytical form for $E[C(f)^2]$, shown in (8.5).

Appendix 2

BLES program overview

A2.1 Introduction

The purpose of this appendix is to provide brief details of the level set optimisation code used to obtain most of the results presented in this thesis. The program is called BLES3 (Bath LEvel Set version 3) and is coded in C and compiled on a machine running Mac OS X (10.5). The appendix is split into two main sections. The first details the input and output of the program, the second provides brief details of the source code.

A2.2 Program input and output

This section provide details and examples of the input file, user input to the terminal and the output files.

A2.2.1 Input file

The input file is a text file, without an extension, that contains only numerical values in a specific order. The following generic example input file defines the order of numerical data.

1.0	Young's modulus (E)
0.3	Poisson's ratio (v)
1.0	Element edge length (h)
100 100	No. elements (x, y)
2	No. initial circular holes
30.0 50.0 5.0	1st hole centre coordinates (x, y) and radius
70.0 50.0 5.0	2nd hole centre coordinates (x, y) and radius
1	No. initial square holes
45.0 55.0 15.0 25.0	1st square hole: min x, max x, min y, max y
1	No. single constrained dofs
100.0 0.0	1st constrained dof coordinates (x, y)
2	1st constrained dof direction: 1 = x, 2 = y
1	No. constrained dof areas
-0.1 0.1 -0.1 100.1	1st constrained dof area: min x, max x, min y, max y
1 0	1st constrained dof area direction (x, y): 1 = constrain 0 = free
1	No. point loads
0.0 100.0	1st point load coordinates (x, y)
10.0 0.25	1st point load mean magnitude and standard deviation
4.712389 0.2	1st point load mean direction and standard deviation (rad)
1	No. uniformly distributed loads
99.9 100.1 24.9 75.1	1st uniform load area: min x, max x, min y, max y
1.0 0.2	1st uniform load mean magnitude per unit length and standard deviation
4.712389 0.2	1st uniform load mean direction and standard deviation

A2.2.2 User terminal input

When the program is run from the terminal the user is asked to input data for some further options. These options are detailed below.

Input prompt	**Details**
Enter name of data file	Name of the input data file. This is also used to generate some output file names.
Enter objective to minimise	Enter 1 to 4 depending on the problem to be solved.
Enter analysis method	1 = BFG (Chapter 4), 2 = AFG (Chapter 3).
Hole insertion?	1 = Enable hole insertion method (Chapter 6), 0 = No hole insertion.
How much info should be printed?	1 = Only output solution data. 2 = Print implicit function and element density output files for each iteration. 3 = Output all data files each iteration (recommended for de-bugging only).
Fix level set function at bc's?	1 = Fix the implicit level set function where dofs are fixed and loads are applied. 0 = Do not fix any part of the implicit function.
Enter combination weight for expected compliance part of objective	This is the value of η in Chapter 8. Only appears if option 4 is selected for the objective.
Enter convergence criterion on objective (x10^-3)	A typical value is between 0.5 and 1.0.
Enter initial maximum number of iterations to perform	Sets an initial limit on the number of iterations. If convergence is not achieved before this limit is reached then the limit can be increased.

A2.2.3 Output files

This section provides a list and details of the output files generated by the program. Files generated in de-bug mode are not included. In the following table, *filename* refers to the name of the input file.

Output file	**Details**
*filename*_NCrd.txt	Coordinates for each node in the fixed square mesh. The three columns are node number, x, y.
PlotShp*K*.txt	Implicit level set function for iteration K. Listed in order of node number.
alpha*K*.txt	Element density file for iteration K. Listed in order of element number.
*filename*_Convergence.txt	Value of objective function and volume for each iteration.
*filename*_lambda.txt	Value of Lagrange multiplier for primary and secondary implicit function for each iteration.

A3.1 Source code

This section provides a brief overview of the source code developed for the BLES3 program. The code uses some FORTRAN 77 code from the HSL library developed by the Numerical Analysis Group at the Rutherford Appleton Laboratory. To compile the C and FORTRAN code into a single program requires a library file to perform the linking, such as libg2c.a or libf2c.a.

File name	**Description**
ABFG.c	Contains functions that determine the type of each element (in, out or approximated), assemble the global stiffness matrix and call sensitivity computation functions from Strain.c.
BCs.c	Contains functions that apply fixed dofs and loading conditions and compute the necessary factors, weights and matrices for the robust optimisation problems (Chapters 7 and 8).
BLES3.c	The main program

Ematrix.c	Contains functions that compute elemental stiffness matrices for the AFG and BFG methods.
FixedGrid.c	Contains functions that interpret or read data from the fixed mesh, such as coordinates of the zero level set.
Levels.c	Contains functions associated with the direct level set implementation (Chapters 5 and 6), including Lagrange multiplier computation, velocity extension, gradient computation and re-initialisation.
Numbering.c	Contains functions that number all nodes and elements and compute nodal coordinates.
Solve.c	Contains functions that call the FORTRAN sub routines, including the multi-frontal solver used to solve the linear finite element equation (Duff 2002).
Strain.c	Contains functions that compute sensitivity values at internal element integration points for the AFG and BFG methods. Also, contains a function to perform the least squares fitting of sensitivity values.
ls_types.h	Contains definitions for the data structures used throughout the program.